

Offert : la plus célèbre des distributions Linux

# DREAM

LE MAGAZINE DE LA MICRO ALTERNATIVE

Mars 1999 - n°60

# La Red Hat 5.2 offerte

- Etape par étape, 50 photos pour bien l'installer
- Intuitif, configurez l'environnement graphique en quelques clics
- Pratique, votre première heure avec Linux



Débutants :
**L'architecture de BeOS expliquée**

Science :
**Initiation au cryptage**

25 pages de programmation :

**Concevez vos propres jeux 3D**

**Tout pour apprendre Java, Perl, Lisp et le C**

**Développez pour X-Window, BeOS, OpenGL...**



Plus fort que le Pentium III :
**l'Alpha 21264**





# Edito

En 1997, nous étions les premiers en France à promouvoir Linux et la micro alternative. Deux ans plus tard, le marché mondial est convaincu de l'utilité du système au pingouin, face à Windows ; en outre, les utilisateurs considèrent en masse que la reconversion dans un système alternatif constitue le dernier cri en matière d'informatique. Même les plus technocrates de nos confrères s'engouffrent dans le sillage contestataire que nous avons dessiné ; à croire qu'il s'agit vraiment là du nouvel Eldorado du silicium. Il n'empêche, le phénomène s'est imposé et vous êtes de plus en plus nombreux à nous rejoindre chaque mois. Pour cette raison, nous vous devons de renouveler notre double exploit : innover et vous divertir. Bienvenue dans la nouvelle formule de Dream...

DR@AM Numéro 60



*Linux est encore une fois la star de ce CD n°60 avec la présence de la RedHat 5.2 pour PC, mais aussi de Kde et du noyau 2.2.1 prêt à être recompilé. La RedHat est devenue probablement la plus connue des distributions Linux. Sa simplicité d'installation et d'utilisation la rend accessible à tous.*

**Amiga**

De nombreux jeux et démos sont présents sur le CD. Vous trouverez également un logiciel éducatif pour apprendre l'anglais, preuve s'il en est que l'Amiga est tout sauf une console de jeux. En prime, nous avons ajouté les derniers patches pour *Linux Power PC*.

**Atari**

Les logiciels choisis sont très variés... D'un côté, les premiers utilitaires pour le Milan, de l'autre la dernière version de *Wensuite*, le tout agrémenté de nombreux autres logiciels... voilà tout ce qui vous attend.

**BeOs**

Profitez des dernières nouveautés pour *BeOS x86* et Power PC. Nous n'avons pas oublié qu'une partie des utilisateurs exploite *BeOs* sur plate-forme Power PC. *BeOs* se veut un système orienté multimédia : qui dit multimédia, dit jeux. Donc, nous vous en offrons.

**Linux**

La première version de *Linux* pour Psion : il s'agit d'une petite merveille à utiliser avec précaution, car le portage est encore trop peu avancé.

**Os/2**

Voici la version 3.3.3 de *Xfree 86*. Cet ensemble assez volumineux permet de réutiliser les logiciels conçus pour *X-Window*.

**RiscOs**

Les machines Acorn disposent elles aussi d'*Unix*. Vous pourrez maintenant profiter de *X11R6.3* pour RiscBSD.
En ce qui concerne RiscOs, la dernière version du pack ArmTex et des démos parmi les plus récentes répondent à l'appel.

**Erratum**

*Vous avez pu remarquer que le dernier CD de Dream proposait des archives de Gnome en grande partie inutilisables. Nous le regrettons. Si la curiosité vous démange, vous pourrez vous pencher sur une version plus ancienne de Gnome (0.20-3) présente dans la distribution RedHat de ce CD. Vous avez également la possibilité de vous rendre sur le site de Gnome (www.gnome.org), ainsi que sur les sites Ftp associés :*
*ftp.gnome.org/pub/GNOME/redhat/latest,*
*ftp://ftp.gnome.org/GNOME*
*et plus près de chez nous,*
*ftp://ftp.fr.gnome.org/pub/gnome*
*Pour nous faire pardonner, nous vous offrirons une version plus récente de Gnome dans le prochain CD de Dream.*

# La RedHat 5.2 pour PC

Le CD contient l'intégralité de la RedHat, à *Netscape* près, complétée par les archives de Kde et du dernier noyau, que les plus courageux pourront essayer d'installer. Cette distribution permet de passer en douceur sous *Linux* avec *Kde* en prime. Ce *Window Manager* reste le plus intuitif, quoique un peu lent, et le plus proche des interfaces classiques que sont *MacOS* ou *Windows 95*. De plus, les nombreux logiciels présents en standard dans la RedHat vous garantissent des heures de plaisir et de découverte. *Linux* est un système jeune, mais il a déjà atteint une pleine maturité, en grande partie grâce au grand nombre de développeurs travaillant à son amélioration. La version x86 de *Linux* reste une des plus stables et des plus abouties disponibles sur le marché. Quant au noyau 2.2.1 de *Linux*, celui-ci transformera de façon radicale votre système préféré. Il cumule de nombreux "plus" : plus stable, plus rapide, il supporte aussi plus de systèmes de fichiers, tout en étant recompilable sur un plus grand nombre de plates-formes.




*Raide mat.*

# KHEOPS




Distributeur français spécialiste de la vente dans les pays francophones des distributions LINUX et des applications optimisées pour ce système d'exploitation, KHEOPS vous propose un ensemble de prestations pour migrer vos applications sous LINUX.
Disposant d'un important réseaux de distributeurs sur les marchés français, belge, suisse et canadien, KHEOPS distribue les produits édités en français par les LOGICIELS DU SOLEIL, RED HAT France, CALDERA, APPLIX et d'autres acteurs du monde LINUX.

**Retrouvez-nous dès à présent sur INTERNET à l'adresse suivante : HTTP://WWW.KHEOPSFRANCE.COM**

Tél. · 01 40 56 91 11 - Fax · 01 40 56 35 34
Siège à Sophia Antipolis
Tél. : 04 92 13 29 56 - Fax : 04 93 73 34 99

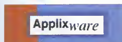
Venez consulter notre site et retrouvez chaque semaine des nouveautés : jeux concours pour gagner des livres O'REILLY, des distributions, des ordinateurs, des bons d'achat...

























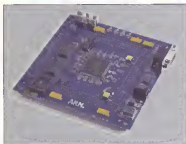




TECHNOLOGIE



# Arm, confirmation d'une année à succès


*La carte d'évaluation AEB-1* d'Arm.

L'année 1998 aura vraiment été l'année Arm (www.arm.com). Le nombre de processeurs à base de technologie Arm fabriqués l'an passé est passé d'un peu plus de 10 millions d'unités à plus de 50 millions d'unités dont une majorité d'*Arm7* (pour un chiffre d'affaires de plus de 400 millions de francs). Les résultats du dernier trimestre de l'année 1998 donnent à penser que la croissance sera encore exceptionnelle cette année.

A titre de comparaison, le nombre de processeurs à base de technologie x86 (Intel, Amd, Cyrix, etc.) sortis des usines en 1998 est de l'ordre de 100 à 120 millions d'unités. Arm commence bien l'année avec ses nouveaux contrats concernant la téléphonie mobile, ainsi que des licences récemment cédées à Toshiba et Samsung. La firme prévoit même de doubler son personnel d'ici 2004. L'*Arm7*, quoique ancien, est au fait des évolutions : VLSI le propose maintenant sous forme de processeur à deux noyaux, un Arm7TDMI (100 MHz) et un OakDSP (80 MHz). Mieux, il est possible d'obtenir des Arm7 gravés en 0.2 microns. Intel pour sa part, annonce un *StrongArm 220*, suivi des *StrongArm 2100* et *2500*, tous axés sur la technologie des *SA110* et *SA1500/1501*. Ceux-ci, prévus pour l'an 2000, fonctionneront à des fréquences allant de 100 à 600 MHz et seront gravés en technologie 0.18 microns. Arm a réussi son pari qui était de devenir le numéro 1 du Risc 32 bits. En dépassant la cap des 120 millions d'unités fabriquées par an, la technologie Arm pourrait devenir la plus utilisée au monde. Un nouveau défi pour la petite société anglaise ?

SERVEURS

# IBM se lance dans Linux

IBM (www.ibm.com) est en train de faire un maximum d'efforts pour assurer un support complet de *Linux* dès le mois de mars. Cette décision assez surprenante de la part de Big Blue fait suite à une étude très sérieuse sur l'impact de *Linux* sur le marché des systèmes d'exploitation. IBM souhaite aussi précharger *Linux* sur ses *RS/6000*, des machines haut de gamme à base de PowerPC et Power (www.rs6000.ibm.com). Effectivement, les premiers essais de *Linux* sur cette plate-forme révèlent des performances surprenantes. En conséquence, IBM est en discussion avec diverses sociétés, dont l'incontournable RedHat et LinuxPPC Inc. La firme est même parvenue à un accord avec RedHat pour la distribution de *Linux* sur divers produits, dont les serveurs Netfinity, mais aussi les *PC300*, IntelliStations et ThinkPads.

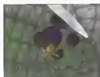

*Un serveur compact RS/6000.*

La concurrence entre IBM et Compaq sur le marché des serveurs *Linux* risque, décidément, d'être rude.

JEU

# Enfin un vrai jeu pour Linux !

Aegis Simulation Technologies propose *BFRIS Zero Gravity Fighter Combat* qui est un jeu de combat en 3D où des vaisseaux s'affrontent dans des espaces clos reliés par des téléporteurs. Le logiciel est, d'un point de vue graphique, assez novateur. L'intérêt principal est de jouer à plusieurs, en réseau : on peut connecter de 2 à 24 (ou 63 sur un réseau local) personnes simultanément pour des parties endiablées. Le moteur 3D nécessite un accélérateur OpenGl hardware (géré via GLX). Actuellement, seules les cartes 3Dfx Voodoo/Voodoo2




*Copies d'écran du jeu.*

/Voodoo Rush ont été testées avec succès sous *Linux*.

BFRIS nécessite donc une configuration musclée : *Linux 2.2.x*, carte graphique avec accélération hardware OpenGl, carte son, *Pentium II* et 64 Mo de Ram restent vivement conseillés. BFRIS est d'ores et déjà disponible pour un prix de 40$. Il existe, en plus de la version *Linux*, une version *Windows (95, 98, NT)*. A ce jour, BFRIS constitue le seul titre ludique avec un pareil niveau d'aboutissement à être disponible sous *Linux*.

TECHNOLOGIE

# L'arrivée des "Be machines" ! (suite)

La couverture du dernier numéro de *Dream* présentait une accroche énigmatique sur des machines multiprocesseurs sous *BeOS* , hélas, point d'article dans le magazine. L'erreur est maintenant réparée.

La *Release 4* de BeOS (www.be.com) constitue une réelle avancée : certains voient dans ce système une alternative viable à *Windows 98*. Si *Linux* s'impose comme un *Windows NT* "killer", *BeOS* représente une solution fiable et facile d'utilisation pour le grand public. *BeOS* s'avère stable, rapide, multitâche et administre parfaitement les machines multiprocesseurs. De surcroît, les possibilités logicielles sont complètes : multimédia, fontes anti-aliasées, support *OpenGl*. *BeOS* fonctionne indifféremment sur PC ou Mac PowerPC. En revanche, le support très limité des cartes d'extensions oblige à construire une machine sur mesure pour ce système. Un nouveau constructeur propose des machines fonctionnant sous *BeOS*. Son nom se montre très évocateur, puisqu'il s'agit de BeMachines (www.bemachines.com). Les ordinateurs sont proposés à des prix raisonnables et offrent la particularité d'être pourvus de un à quatre microprocesseurs. A noter : les configurations proposées sont très bien équipées, ce qui promet des performances sans commune mesure avec celles des PC classiques.

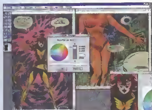


*Les premières Be Machines.*

APPLICATION

## Vision 3.5

Une nouvelle version de ce très étonnant logiciel de dessin vient d'arriver. La gestion des cartes graphiques s'est encore améliorée, ce qui représente une belle initiative à l'heure du Milan. Rappelons qu'en dehors de toutes ses qualités de logiciel de dessin, *Vision 3.5* symbolise le meilleur catalogueur d'images sur Atari. A ce titre, il s'avère absolument indispensable pour tout infographiste sur ces machines. Le prix de *Vision 3.5* est de 180 francs, et la mise à jour de 50 francs. A ce tarif, il serait fort difficile de s'en passer, ne croyez-vous pas ? (www.multimania.com/jlusetti/vision.htm)



REVOLUTION

## L'Amulet 3 arrive en juin

L'Amulet (www.cs.man.ac.uk/amulet/) sera le premier processeur asynchrone (sans horloge) construit de façon industrielle. Il est développé par l'université de Manchester et repose sur un noyau Arm. Son arrivée est prévue pour juin et une de ses utilisations principales sera l'intégration dans des produits de téléphonie mobile.



*Vue interne de l'Amulet 1.*

L'Amulet introduit un nouveau concept dans les processeurs de faible consommation, puisqu'en cas d'inactivité, sa consommation devient quasiment nulle : alors que les processeurs tournent à des fréquences fixes, l'Amulet ne s'anime que lorsqu'on s'en sert... Simple, mais révolutionnaire.

**• L'avenir de BeOS sur Metabox et Power PC**

*Jean Louis Gassée renouvelle son soutien à la communauté Power PC et aux machines de Metabox Infonet AG (ex PIOS Computer AG). Motorola semble moins intéressé qu'Intel pour l'évolution de BeOs ce qui explique en partie que Be Inc se penche plus volontiers vers les PC. Metabox Infonet souhaite vivement travailler avec Be sur l'adaptation de BeOs sur leur machine. Jean Louis Gassée n'y voit aucune objection, mais ne peut s'y investir.*

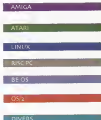
**• Linux supporte l'Euro**

*L'Euro Pack, dans sa version 1.5 bêta 1, permet à Linux de supporter les fontes latin0 (ISO-8859-15 latin9). Le package comprend des patches et des documentations, réservés à l'utilisation du symbole Euro sur Linux. L'Euro Pack se trouve disponible sur les sites suivants : ftp://jaguar.nfrance.com/pub/linux/french/EURO-1.5-beta1.tar.bz2 ftp://amandine.nfrance.com/pub/linux/french/EURO-1.5-beta1.tar.bz2*

**• Du nouveau chez Symbian**

*Symbian (www.symbian.com) regroupe actuellement Psion, Nokia, Ericsson et Motorola. Un des buts de ce groupe est de freiner la progression de Windows Ce et d'asseoir Epoc32, le système de Psion.*

*Sun Microsystems a décidé de rejoindre Symbian qu'il perçoit comme une "merveilleuse" société. Sony cherche également à faire partie de Symbian et Siemens se dit intéressé. Ces bonnes nouvelles devraient contribuer au succès de Symbian.*



• **Compaq montre son attachement à Linux**
*Compaq présente une page sur le hardware des Alpha et son intégration sous Linux. C'est la première fois qu'un groupe informatique de cette taille parle officiellement de Linux dans ses pages. L'exemple de Compaq risque d'être rapidement suivi par d'autres fabricants d'ordinateurs. (www.unix digital.com/linux/)*

• **IBM construit des processeurs Alpha**
*Les rumeurs annoncent qu'IBM, en accord avec Compaq, semble être décidé à fabriquer des processeurs Alpha dans ses usines. AMD pourrait aussi rentrer dans la danse, lorsque son usine de Dresde en Allemagne sera prête.*

RESEAU

# Le réseau danse la Samba

Samba (www.samba.org et lwn.net/1999/0121/samba.html) est un serveur de fichiers et d'impressions gratuit, fonctionnant sur des machines *Unix* et destiné à accueillir pour clients des machines sous *Windows*. La nouvelle version *2.0* de ce logiciel constitue le plus rapide de serveur au monde pour ce type d'utilisation. L'ajout de Samba Web Administration Tool (SWAT) permet de configurer facilement Samba via un navigateur Internet situé sur n'importe quel client. Grâce à toutes ces qualités, ce logiciel a été adopté par les gens de chez Silicon Graphics, ainsi que pour de nombreux "Thin Servers" comme le NetWinder de chez Hardware Canada Computing (appareil anciennement développé par Corel). Samba est entièrement écrit selon les spécificités de la norme Posix et se voit livré avec tous ses sources. Il y a donc moyen de le recompiler sur de nombreuses plates-formes *Unix*.



*Réseau de PC.*

AGRESSION

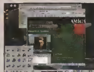
# Sun fustige (encore) ses adversaires

Après avoir reproché à Silicon Graphics d'avoir développé une gamme de PC, Sun (re-)tire à boulets rouges sur Microsoft et *Windows 2000*, qu'il renomme Win-DOZE 2000 ou encore "*Windows sometime around 2000*" (www.sun.com/reality-check/headsup990202.html). Bref, Sun essaie encore de se présenter en tant que sauveur et comme le meneur du monde alternatif. Malheureusement, le côté mercantile refait vite surface : le mois dernier, sa critique servait de publicité pour ses stations Ultra 5 et 10. Ce mois-ci, Solaris prend Sun pour cible dans son pamphlet (rappel : Solaris fonctionne aussi sur plate-forme x86).

## La galerie du mois

***Le groupe AmigaOS.com s'est attaché à déterminer les fonctionnalités du futur Amiga OS 5. Il propose, entre autres, les nouveautés qui devraient l'agrémenter. Leur site www.amigaos.com démontre que les bureaux de style ne constituent pas l'apanage des utilisateurs d'X-Window ou de MacOS... Admirez !***











EXTENSION

# Nouvelle carte Centurbo II pour Falcon

Centek (www.centek.fr) propose une nouvelle version de sa carte Centurbo II, la Centurbo IIb, qui est disponible depuis février. Celle-ci comporte un *68030* à 50 MHz et pousse le bus et le Dsp à 50 MHz. Elle permet également d'étendre la mémoire de votre Falcon de 128 Mo de Ram Edo et d'ajouter un coprocesseur arithmétique, cadencé lui aussi à 50 MHz. Les différences entre la Centurbo II et la IIb portent sur la correction de plusieurs problèmes concernant le port série et le coprocesseur. Le gestionnaire de mémoire se révèle aussi plus performant. Cette carte s'installe par le biais d'un connecteur et sans aucune soudure, ce qui offre un avantage supplémentaire. Le prix annoncé est d'environ 1890 francs TTC. Centek devrait également bientôt se trouver en mesure de fournir le Phenix rev 1... Il s'agit d'une version modifiée par rapport aux spécifications de base, avec suppression du Quicc au profit d'une architecture Pci, multiples Dsp sur carte Pci et mémoire Sdram.




*Carte Centurbo IIb.*

SYSTÈME

## EROS : The Extremely Reliable Operating System

*Eros* (www.cis.upenn.edu/~eros/) est un nouveau système d'exploitation rapide et sécurisé, développé par l'Université de Pennsylvanie. Il s'agit avant tout d'un projet de chercheurs. On le dit trop peu souvent, mais il existe des centaines de systèmes d'exploitation viables dans le monde informatique. Nombre d'entre eux fonctionnent sur plate-forme *x86*. De temps à autre, un de ces systèmes inconnus passe sous les feux de la rampe. Hier, *Qnx* bénéficiait d'un vedettanat soudain et aujourd'hui, c'est au tour d'*Eros*.

Les sources de la pre-release d'*Eros* sont disponibles et recompilables sous tout type d'*Unix*. *Eros* fonctionne sur des machines de faible capacité : 20 Mo d'espace libre sur le disque, 640 Ko de mémoire et un *486* suffiront pour utiliser ce système, même s'il vaut mieux avoir une configuration plus musclée. *Eros* ne fournit pas d'interface graphique, bien que cela soit prévu. Le système devrait rapidement être compatible Posix et Java.

Cupidon, le symbole d'*Eros*

Il ne faut pas se tromper, *Eros* est d'une conception radicalement différente d'*Unix*. Il descend de *KeyKOS*, qui lui-même reposait sur *Gnosis*. Les développeurs prévoient une taille finale pour le noyau de 50 à 60 Ko. *Eros* est gratuit pour un usage privé, mais pas Open Source. Jonathan S. Shapiro ne prévoit de passer son système en Open Source que si ce modèle fait commercialement ses preuves.

EMULATION

## Nouvelle version d'Amiga Forever

Cloanto (www.cloanto.com/amiga/forever/) propose une nouvelle version de son pack Amiga Forever. Cette release comprend de nombreux logiciels intégrés, tels *Amiga Explorer 3.0*, *Personal Paint 7.1*, *TurboText*, *KingCON*, *ToolManager 5* et un utilitaire Workbench pour gérer les archives par simple tirer/lâcher avec la souris. La version online propose d'office ces améliorations ; en revanche, il faudra attendre un petit peu pour le CD-Rom correctif. Les utilisateurs enregistrés de la version online bénéficient gratuitement de cette mise à jour : la taille de l'archive, comprenant l'ensemble des logiciels, ne fait que 6.5 Mo. Amiga Forever est proposé au prix d'environ 30$ pour la version online et de 50$ pour la version CD-Rom ; il s'agit un prix raisonnable pour profiter de l'un des meilleurs systèmes d'exploitation multitâche du marché.

Vous pourrez trouver d'autres renseignements à l'adresse suivante : www.cucug.org/amiga/aminews/1999/990131-cloanto.html

Damier, la malédiction d'Amiga ?

**• Le TrueType sous X-Window !**

*Toutes ces adresses pointent vers un seul applicatif : les binaires de Xfree 86, avec support pour les fontes TrueType... A n'en pas douter, il s'agit là d'un grand pas pour les utilisateurs Linux issus du monde Windows.*

*www.dcs.ed.ac.uk/home/jec/programs/xfsft/*

*www.darmstadt.gmd.de/~pommnitz/xfsft.html*

*www.darmstadt.gmd.de/~pommnitz/XF86-xfsft/index.html*

*www.microsoft.com/typography/fontpack/default.htm*

*Ying Zhang propose également des pages d'aide pour certains éléments de Linux, dont une partie sur l'utilisation des polices TrueType dans X-Window (www.sfu.ca/~yzhang/linux/).*

**• 3 Go de logiciels pour OS/2**

*Hobbes (hobbes.nmsu.edu) a franchi le cap des 3 Go de logiciels pour OS/2. Le site est encore en pleine croissance et les gestionnaires se fixent une nouvelle limite de 4 Go. OS/2 n'est donc définitivement pas prêt de mourir. Le site Hobbes est rapide mais très chargé. Heureusement, il existe de nombreux sites miroirs. Leur liste peut être chargée à l'adresse ftp://hobbes.nmsu.edu/0mirrors.txt.*

**• SGI modifie son plan d'action**

*Nous nous sommes déjà fait l'écho de la volonté de Silicon Graphics Inc. qui désirait créer des stations haut de gamme à base de Pentium II et fonctionnant sous Windows NT. SGI annonce maintenant un autre produit, un serveur, toujours à base de Pentium II, mais cette fois-ci deux systèmes d'exploitation sont officiellement supportés : Windows NT et Linux.*

**• Inferno gratuit !**

*Lucent Technologies (www.lucent-inferno.com) propose de télécharger la version 2.3 Alpha de son système d'exploitation temps réel, Inferno. Le kit de développement comprend des exécutables pour Windows, mais aussi pour divers Unix, dont Linux.*

**• Faites-vous rembourser Windows !**

*Tel était le mot d'ordre pour une journée de protestation (le 15 février dernier) contre le système de Microsoft et pour les environnements libres. Peu de personnes ont demandé à être remboursées, et aucune ne l'a été. Effectivement, se faire rembourser Windows est une opération laborieuse : un utilisateur australien a mis quatre mois pour convaincre Toshiba de lui reprendre Windows. Malgré tout, le message semble avoir été bien compris. Affaire à suivre. (www.LinuxMall.com/refund/)*

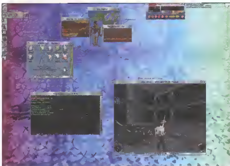
**• Nouveau point d'entrée pour les systèmes alternatifs**

*Peanuts-Archive vient de mettre en place une page destinée à faciliter la recherche de logiciels pour NextStep, OpenStep, MacOs X et WebObjects. L'index du moteur est réactualisé chaque jour. www.peanuts.org/Search.html*

REORGANISATION

# Amiga : Gateway remet tout à plat !

C'en était trop : entre les accords pour le hardware qui n'aboutissent pas et le développement logiciel qui n'avance pas d'un octet, le constructeur Gateway a tapé du poing sur la table au sujet du dossier Amiga ! Au sommaire de cette véhémente remise en question : une réorganisation complète des filiales en charge de la marque et des délais plus impératifs que jamais. Désormais, il y aura une société mère, Amiga Corp. (basée à San Diego), avec un patron unique, Jim Colas, le vice-président même de Gateway. Jeff Schindler et Petro Tyschtschenko se retrouvent quant à eux directeur du marketing et directeur administratif dans leurs sous-filiales respectives (Amiga Inc. pour le premier et Amiga International pour le second). Promotion également pour Allan Havemose, lequel se retrouve vice-président pour le développement logiciel. Installée dans la Silicon Valley, son équipe est chargée de développer une enveloppe "AmigaOS-like" autour du noyau Neutrino de QNX. Les objectifs de Jim Colas sont clairs : "*nous sortirons avant la fin de l'année et dans le monde entier un ordinateur personnel dans la tradition de l'Amiga 500, une station haut de gamme pour le développement et un boîtier pour naviguer sur Internet depuis sa télé*". Par ailleurs, la mise à jour du système actuel pour la gamme "Classic" (le tant attendu *AmigaOS 3.5*) devrait être disponible au World of Amiga 99. A noter que Jim Collas était auparavant le représentant de Gateway lors de chaque conférence de presse ou manifestation Amiga.



*Le futur AmigaOS ?*

ANNIVERSAIRES

# Festivités dans le monde Acorn

L'Acorn News Service (ANS), vient de souffler sa première bougie. Cette lettre d'information, beaucoup critiquée à sa sortie, est devenue un document essentiel pour s'informer sur l'actualité Risc Os. Vous trouverez plus de détails sur le site www.acornusers.org/ans/.

Acorn Arcade a également un an. Ce magnifique site regroupe des informations de premier choix sur les jeux et démos pour plate-forme Acorn. Voilà un point d'entrée intéressant pour les fans de jeux (www.acornarcade.com).



*Firestorter, le prochain hit du Risc Pc ?*

PIRATAGE

## Honni soit qui mal y pense...

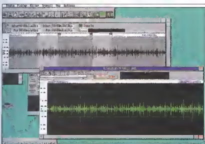


Nous avons fait état le mois dernier d'un CD-Rom rassemblant la quasi-totalité des Diskmags Atari. Celui-ci aurait dû normalement se révéler très intéressant, puisqu'il avait l'intention de rassembler des e-zines comme l'*UnderCover Mag* ou le célèbre *ST News*. Après enquête, il semble que ce CD-Rom ne soit rien d'autre qu'un énième produit pirate, qui offre la totalité des disquettes de *ST Magazine* ou *Start Micro*. Or, aucun accord n'a été passé entre les sociétés de ces deux anciens magazines et l'auteur du CD-Rom. Et la propriété intellectuelle, alors ? Donc, possesseurs de ST ou de Falcon, si vous recevez une publicité proposant de vous vendre ce CD, jetez-la. Ce n'est pas parce qu'Atari n'existe plus qu'il faut faire n'importe quoi !

APPLICATION

## V Trax sur Atari

Vous connaissez tous *Studio Son*, ce célébrissime programme de montage audionumérique, initialement écrit pour Falcon. Celui-ci fut conçu au départ en version deux pistes, puis annoncé en huit pistes. Un logiciel de montage n'ayant pas la même finalité ni la même conception qu'un multipistes, ses auteurs (Cédric Ricard et David René Loiseau) ont décidé de le scinder en deux programmes différents : *Studio Son* pour le montage et *V Trax* pour le multipistes. Une toute nouvelle version de *V Trax* a fait son apparition début janvier. Elle gère la fast Ram de la Centurbo II et rajoute de nouvelles fonctions. Le site de Cédric Ricard (www.emi.u-bordeaux.fr/~ricard/) mentionne qu'il ne reste plus aucun bug connu, du moins à ses yeux. Par conséquent, si vous en trouvez, signalez-les lui, afin qu'il les corrige dans sa prochaine mouture.



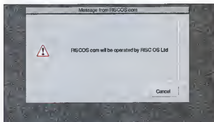
*Le logiciel Studio Son.*

**• Achetez vos logiciels BeOS... en France.** *Studio Capitale (www.studiocapitale.com) propose la revente de produits pour BeOS. Cela va du système lui-même, en passant par des outils, tel l'intégré Productive (traitement de textes, tableur, graphiques, présentation www.gobe.com) ou encore Mail It. Cette société est actuellement une des seules en France à offrir un tel service dans le domaine du logiciel pour BeOS.*

RENOUVEAU

# Bonnes nouvelles pour le monde Acorn

Castle technology (www.castle-technology.co.uk) reprend à son compte la fabrication des *A7000+*. Les stocks seront disponibles dès le mois de mars. Du côté des logiciels, RiscOs Ltd (www.riscos.com) annonce la création de la fondation RiscOs, destinée à favoriser le développement de ce système d'exploitation. Ce nouveau groupe remplacera l'actuel Clan dans son action de promotion et enverra à ses membres un bulletin trimestriel ainsi que deux CD par an. Le premier projet est de fournir une version 32 bits de Risc Os pour le nouveau millénaire. L'actuel Risc Os, grâce à son mode 26 bits, se voit cantonné aux *Arm 6* et *7*, ainsi qu'aux *StrongArm 110* et *120*. L'utilisation du mode 32 bits permet au système d'être supporté par toute la famille des processeurs Arm : *Arm 9* puis Amulet et *Arm 10*. Pour adhérer à la fondation, il faut verser (une seule fois) une somme allant de 250 à 300 francs.




*A vos Risc OS et périls.*

## Droit de réponse

***L'éditeur ADFI a réagi à la suite de notre test du logiciel de dessin vectoriel DrawStudio pour Amiga. Nous leur laissons la parole : "La société A.D.F.I. est le distributeur exclusif pour tous les pays francophones du logiciel de dessin et déformation d'images DrawStudio testé dans le numéro 58 de Dream (janvier 1999). Nous avons noté quelques erreurs qu'il convient de corriger.***

***• DrawStudio est un logiciel commercial. Il ne s'agit pas d'un "shareware" (logiciel à paiement compensé). La confusion du testeur provient peut-être du fait qu'il existe une version de démonstration fort limitée ; l'utilisateur de celle-ci peut alors acquérir la vraie version en payant les droits d'auteur.***

***• DrawStudio est destiné à fonctionner de préférence avec une carte graphique. Sinon, à défaut d'une telle carte, on peut l'utiliser efficacement sur un écran standard Multiscan de 16 à 256 couleurs. Le testeur a rencontré quelques problèmes, mais il aurait pu lire la documentation car il y aurait trouvé la réponse : si des problèmes de plantages bien réels ont parfois été décelés, ils proviennent en réalité de l'utilisation conjointe de commodités mal programmées et étrangères au logiciel. Il suffit de les enlever pour que tout revienne à la normale.***

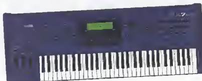
***• Le testeur a découvert une interface «gris mélasse et rose bonbon». Un logiciel comme DrawStudio est destiné à être utilisé en vraies couleurs et sur un écran de qualité. A défaut, puisque DrawStudio utilise M.U.I., sur les seuls modes télévisuels Pal et Ntsc pris en 16 ou 32 couleurs, quelques crayons non fixés peuvent se retrouver dans une couleur inadéquate. Pour résoudre cette difficulté, il suffit de les fixer avec diverses commodités du domaine public, ou d'utiliser plus de 32 couleurs sur les modes incriminés, ou enfin d'utiliser une carte graphique. C'est dans la documentation.***

***• Le logiciel évolue en permanence. Les textes de l'interface sont susceptibles d'être modifiés selon les remarques des utilisateurs. Dream nous a demandé de recevoir en urgence le logiciel pour écrire un article : nous avons alors fourni la toute dernière version 2 ainsi que le fac-similé du projet du nouveau mode d'emploi. Dans ces conditions d'urgence, il était donc normal que les termes de ce fac-similé diffèrent légèrement de ceux du logiciel. Nous avons été quelques peu surpris que cela soit mis en exergue par l'auteur du test car nous avions signalé tout cela par téléphone et par écrit au magazine".***

MUSIQUE

# Be fait le forcing dans le domaine musical

Be (www.be.com) a annoncé début février son arrivée en force dans le monde de l'informatique musicale. La société a négocié des accords avec pas moins de 26 groupes de développeurs pour assurer la sortie de logiciels musicaux sous *BeOS* d'ici à trois mois. On retrouve tous les grands noms de ce domaine, de Yahama à Emu en passant par Steinberg (Cubase), mais aussi Capitale pour un portage de ses logiciels Atari, *Studio Son* et *V Tracks*. Il est curieux de noter combien la politique marketing de Be est proche de celle des fabricants de consoles de jeux. Be s'assure non seulement de la disponibilité de son OS, mais donne surtout la priorité à l'offre logicielle. *NextStep* avait usé, à sa sortie, de la même tactique. *BeOS* est une plate-forme très performante pour les applications multimédias. La musique pourrait donc devenir une des pierres angulaires du système Be. Dans la plupart des projets en cours de développement, l'usage des connexions inter-programmes est immodéré. Cela combiné à l'interface facile d'utilisation de Be, devrait apporter de nouvelles perspectives concernant les capacités et l'ergonomie de ce type de logiciels, ce qui ne manquera pas de favoriser la créativité.



*Be nouveau ténor du monde de la musique ?*

3D

## Silicon Graphics Inc. aide la communauté Open Source

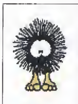



*OpenGL, la perfection 3D.*

*GLX* est une interface permettant de lier *X-Window* et les applications *OpenGl*. Son usage est obligatoire pour porter les applications *OpenGl* sous X. La mise à disposition des sources de *GLX* par SGI offre enfin la possibilité de créer une ouverture vers *OpenGl* pour les systèmes libres que sont *Linux*, *BSD*, et l'ensemble des OS disposant de *X-Window*. Les drivers avec accélération 3D pour *Xfree 86*, attendus pour cette année, ajouteront encore de l'intérêt à ce programme.
(www.xwindows.org
& www.sgi.com/software/opensource/glx/)

SITE

## Site Atari.org



*Petit logo du site Atari.org*

Le site Atari.org a pris une taille importante en un temps assez court. Il faut saluer la performance des volontaires travaillant sur ce projet. Atari.org regroupe un service d'E-mail et de sous-nom de domaine, un service d'hébergement, un moteur de recherche et une base de données regroupant les annonces Atari. L'ensemble de ces services est gratuit. Voici, pour information, le statut d'avancement du site tel qu'il apparaît au moment où ces lignes sont écrites :

Pathfinder 95 %
Free Subdomains 100 % (plus de 200 déjà disponibles !)
News Pages - 80 %
Free WebSite Hosting 80 %
Active Team Members 8
www.atari.org







**• Un bon point pour Sun**
*Sun met le code source de son Java 2 SDK à la disposition des développeurs à travers la licence Community Source. Les développeurs sont donc libres de faire évoluer le SDK selon leur bon plaisir, ou de l'adapter à d'autres plates-formes. Le paiement de royalties à Sun ne se fera que sur les ventes de produits découlant de l'utilisation de ce code. (www.sun.com/software/community source/java2)*

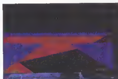
**• Intel : annonces et démentis**
*Après Compaq (avec l'Alpha) et Motorola (Power PC), Intel annonce des puces à 1 GHz. Les annonces (avec dates à la clé) semblent précéder les démentis... Bref, comme d'habitude, la désinformation est de rigueur. La barrière des 1 GHz semble plus que jamais aussi psychologique que celle de l'an 2000.*

**• Un nouveau Communicator pour OS/2**
*Netscape Communicator 4.04 vient d'être mis à jour sur le site IBM Software Choice (www.software.ibm.com/swchoice). Cette version très spéciale de Communicator se situe entre la 4.0x classique et la 4.5 actuellement disponible sur les autres systèmes. La mise à jour concerne un meilleur support de la version 1.1.7 de Java, la gestion des écrans 16 couleurs et la correction des bugs rencontrés dans la première mouture.*

**• Erratum**
*En page 10 du précédent numéro de Dream, il y avait deux erreurs. La première concernait Acorn, qui a revendu ses parts d'Xemplar à Apple (et non pas Acorn). De plus, il manquait la fin de l'article... Il fallait lire "Il ne manquerait plus que l'accord d'Acorn pour que cette extension voie le jour"*

APPLICATION

# Une JVM gratuite pour Mozilla

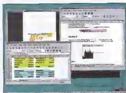


*Un jeu 3D en Java.*

ElectricalFire est le nom d'une Java Virtual Machine entièrement gratuite et développée par les membres du groupe Mozilla (www.mozilla.org). Elle s'ajoute aux nombreux projets déjà existants, à savoir Kaffe, Japhar, TYA et Gnu Classpath. ElectricalFire constitue une JVM améliorée, qui se présente sous la forme d'un compilateur Just In Time, disponible sous *Linux* et *Windows*. Conçue à l'origine pour devenir la JVM de Netscape, cette application s'est finalement changée en projet sous licence NPL (Netscape Public Licence). Le code se veut portable, ce qui laisse à penser que son usage pourra rapidement se généraliser.

APPLICATION

# Siag Office 3.1.6



*Copie d'écran de Siag Office.*

La dernière version de *Siag Office* est disponible. Ce programme est une suite bureautique gratuite comprenant un traitement de texte, un tableur et un logiciel de présentation. Cet ensemble ne peut encore rivaliser avec les logiciels commerciaux, mais son développement est rapide et il propose déjà des fonctionnalités novatrices. Sa grande faiblesse réside dans l'incapacité d'importer et d'exporter les formats de fichiers utilisés habituellement dans ce type de logiciel. *Siag* est aisément extensible, via des plug-ins ou à travers divers langages de programmation. Il est pour sa part programmé en Scheme (www.edu.stockholm.se/siag/).

ECLAIRCISSEMENT

# Jini VS Jene



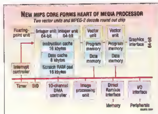

Nous avons parlé le mois dernier de Jini, la technologie de Sun qui prend la forme d'une surcouche de Java et permet l'accès aux appareils de tous les jours. Après quelques recherches, on peut s'apercevoir que Sun manque quelque peu d'imagination dans le choix du nom de ses produits. Insigna Solutions (www.insignia.com/embedded/) a lancé un produit du nom de Jene en septembre 1998. Il s'agit d'une JVM de petite taille et de hautes performances, destinée à l'électronique embarquée : de fait, Jene signifie Java Environment for Embedded. Jene réduit les problèmes inhérents à Java : lenteur, taille et manque de possibilités temps réel. La machine virtuelle utilise un procédé de compilation dynamique, plus performant que les techniques utilisées par les compilateurs "just in time". Jene fonctionne de surcroît sur un grand nombre de processeurs.

JENE™

MATERIEL

# Sony va de l'avant

Après avoir vendu près de 50 millions de sa PlayStation, Sony laisse enfin entrevoir l'arrivée d'un nouveau produit. Un processeur 128 bits co-développé par Toshiba et Sony prendrait place dans la nouvelle console. Celui-ci, de technologie Mips et gravé à 0.18 microns, comprendrait trois unités de calcul et fournirait 5GFlops de puissance brute : de quoi traiter plus de 30 millions de polygones par seconde. Le processeur apportera aussi des fonctions de traitement graphique, dont un décodeur Mpeg2. Cette petite merveille tournera à 250 MHz pour une fréquence de bus de 125 MHz.

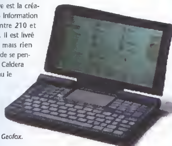



*Un jeu 3D en Java.*

EMULATION

## Les applications DOS sous Epoc32

Les machines sous *Epoc32* (*Psion Series 5* et Geofox) disposent maintenant d'un émulateur PC/XT capable de simuler un *80186* et les spécificités des *HP200LX*. Une émulation PC est intéressante à plus d'un titre : des milliers de logiciels étaient et sont encore disponibles sous *DOS*, ce qui agrandit instantanément la liste des logiciels utilisables sur Psion. Le fait de supporter les applications écrites pour le *HP200LX* donne la possibilité de profiter de logiciels conçus pour une machine relativement récente, dont les capacités graphiques sont proches du Psion. Cet utilitaire est la création de NB Information et coûte entre 210 et 300 francs. Il est livré sans *DOS*, mais rien n'empêche de se pencher sur le Caldera OpenDos ou le FreeDos. (www.nbinfo.co.uk)



*Le Geofox.*

SYSTEME

## Quake II et Pentium III sous BeOS

Le 19 février dernier, Intel a fait une démonstration du nouveau *Pentium III* en utilisant *BeOS R4.1* et ses nouvelles fonctions d'accélération hardware *Open Gl*. La machine était équipée d'une carte Voodoo II et le logiciel utilisé pour la démonstration était *Quake II*. La version *4.1* de *BeOS* est attendue pour la fin du premier trimestre. Quant à *Quake II*, tout porte à croire qu'il sera disponible rapidement, après l'arrivée du nouveau système. Concernant la sortie de *Quake III* sous *BeOS*, John Carmack affirme que l'ID Software fera probablement appel à des programmeurs indépendants pour assurer le portage.



*Une tranche de Quake ?*

Be est devenu un développeur enregistré pour le *Pentium III*, alors même que certains voient dans ce processeur un échec annoncé : ils reprochent au *Pentium III* d'être trop cher et trop peu innovant (surtout face au tout récent *K6 III* d'AMD).

http://developer.intel.com/drg/pentiumiii/tools/be.htm

www.benews.com/story/?ID=666

Nota : vous aurez noté que l'ID de la news de chez BeNews est assez amusant pour une annonce concernant *Quake* !

# Alpha 21264, le tueur de Pentium

*Le microprocesseur Alpha 21264 symbolise la troisième génération de l'architecture Alpha, apparue le 25 février 1992. Aujourd'hui, les premières machines équipées du 21264 commencent à apparaître. Mais quelles sont donc les caractéristiques de ce processeur révolutionnaire ?*

L'histoire du microprocesseur Alpha débuta pendant l'été 1988, dans un laboratoire de Digital. Plusieurs ingénieurs commencèrent à envisager qu'il fallait donner un successeur au VAX, menacé par de nouveaux concurrents RISC : le SPARC de Sun, le PA-RISC de HP ou encore le RISC de MIPS, affichaient tous des performances meilleures que les siennes, pour un coût moindre. Il y avait alors une croissance de la demande pour des stations de travail de plus en plus puissantes, un des marchés traditionnels du VAX. Les microprocesseurs équipant les micro-ordinateurs devenaient eux aussi de plus en plus rapides, et le remplacement des terminaux passifs, habituellement connectés aux VAX par des PC, devenait fréquent. Il était alors permis de penser que ce phénomène allait se poursuivre et que les micro-ordinateurs deviendraient prochainement plus prompts que les grosses machines. L'objectif du groupe de travail formé pendant cet été 1988 était donc de trouver des solutions, pour faire évoluer l'architecture VAX. L'une d'entre elles consistait à le transformer en un microprocesseur RISC. Ce groupe de travail s'appela EVAX. Cette étiquette fut ensuite utilisée pour définir les noms de code des versions successives du processeur Alpha.

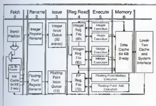



**Le premier Alpha, le 21064**

La génération de départ prit la forme du *EV4*. Le 4 correspondait à la quatrième phase de la technologie CMOS du processus de fabrication. Cette technologie se caractérisait par un pas de grille de 0,75 microns et par trois niveaux de métal (voir encadré). Le processeur qui en découla s'appela l'Alpha *AXP 21064*, le 21 renvoyant au 21ème siècle, le 64 aux 64 bits du processeur et le 0 central au numéro de génération. Il s'agit alors du premier microprocesseur 64 bits de l'Histoire. A l'époque, Digital ne put déposer la marque Alpha, ce qui explique l'ajout du terme AXP, lequel n'a pas de signification particulière. Celui-ci a aujourd'hui disparu. Annoncé le 25 février 1992, ce microprocesseur était alors le plus rapide du marché. Les fréquences initialement disponibles atteignaient 150, 166 et 200 MHz. Le *21064* a ensuite été remplacé par le *21064A* (*EV45*) le 18 octobre 1993. Celui-ci conservait la même architecture, mais se voyait fabriqué en technologie 0,5 microns, avec quatre niveaux de métal, ce qui autorisait des fréquences de 225 et 275 MHz. Le potentiel de transistors supplémentaires a été exploité pour augmenter la taille des caches instructions et données, de 8 à 16 Ko chacun. Le nombre de transistors passa de 1,67 millions à 2,8 millions.

**Le 21164, la deuxième génération**

Le *21164* (aussi appelé *EV5*) constitue la première évolution importante de l'architecture Alpha. Apparu le 7 septembre 1994, il se caractérise par un nombre de transistors plus élevé (9,3 millions) et la possibilité d'exécuter jusqu'à quatre instructions par cycle, alors que le *21064* ne pouvait en exécuter que deux. La technologie de fabrication, identique à celle du *21064A*, n'autorise que des fréquences à peine plus élevées (266 et 300 MHz), mais son architecture interne lui permet de franchir le premier la barre du BIPS (milliard d'instructions par seconde, le milliard se disant billion en anglais). Son cache interne de niveau 1 est réduit à 8 Ko pour les instructions et 8 Ko pour les données, mais la puce comporte 96 Ko de cache de niveau 2 directement intégrés. La réduction du cache primaire se rattache à la nécessité de pouvoir y accéder en un seul cycle (soit 3,3 nanosecondes à 300 MHz). La vitesse et la taille du processeur sont en effet telles qu'un cache plus grand aurait recquis deux cycles pour se trouver en mesure d'en faire autant. L'évolution naturelle du *21164* a débouché sur le processeur *EV56*, qui se caractérise par un doublement de la fréquence (jusqu'à 600 MHz). Il a été annoncé le 31 mars 1997. Le gain de performances obtenu par l'augmentation de la fréquence s'avère très significatif, puisque ce processeur atteint 2,4 Bips soit 18,4 SPECint95 et 21,3 SPECfp95.




*Principe de fonctionnement.*



## Dream prof

**• SPECint95 et SPECfp95**

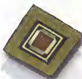
*SPEC (http://www.specbench.org) est un organisme qui a défini un ensemble de tests permettant de mesurer la performance brute des machines en calcul entier (SPECint) et flottant (SPECfp). La version actuelle des tests a été définie en 1995. Enfin, précisons que cette structure ne dépend aucunement des constructeurs.*

**Le dernier-né, le 21264**

Le *21264* (nom de code *EV6*) représente la génération actuelle du processeur Alpha. Dévoilé le 24 octobre 1996 lors du Microprocessor Forum (manifestation annuelle pendant laquelle les autorités compétentes présentent la plupart des recherches dans le domaine des processeurs), il n'a réellement été annoncé que le 2 février 1998. Les premières machines Digital à l'exploiter ont commencé à apparaître à la fin 1998. Des cartes mères compatibles se trouvent aussi disponibles depuis peu. Sa technologie de fabrication utilise un pas de grille de 0,35 microns avec six niveaux de métal. Il compte 15,2 millions de transistors ; seulement six d'entre eux se voient affectés au processeur lui-même. Les autres transistors correspondent aux caches et aux tables de prédiction des branchements. La première version est disponible en 575 MHz, mais ce processeur entend atteindre les 800 MHz dès cette année, grâce à la technologie 0,25 microns, puis un Gigahertz, à cause de la technologie 0,18 microns (on réserve celle-ci pour le processeur *EV7*) que ses fabricants lui attribueront l'année prochaine. Sa puissance brute est de 30,3 SPECint95 et de 47,7 SPECfp95, ce qui signifie qu'elle se montre une fois et demie à deux fois plus rapide que ses concurrents actuels (voir tableau). Il atteint ces performances grâce à plusieurs améliorations significatives (voir schéma).

**Les caractéristiques du 21264**

Jusqu'à six instructions peuvent être exécutées en parallèle, soit quatre entières et deux flottantes. Dans la pratique, quatre instructions bénéficient d'un traitement simultané. Le processeur dispose de deux fois quatre-vingts registres entiers et de quatre-vingts registres flottants sachant que l'architecture Alpha définit en fait trente-deux registres de chaque type – pour pouvoir s'acquitter au même moment de toutes ces tâches. Les registres supplémentaires sont rebaptisés dynamiquement en fonction des besoins, et les unités de calcul, spécialisées. On dispose ainsi de deux unités entières consacrées aux accès mémoire (qu'il s'agisse de données entières ou flottantes). Cependant, elles peuvent également effectuer des additions, tandis que les deux autres unités de calcul entier s'utilisent pour toutes les autres opérations. Celles-ci se distinguent malgré tout par une petite spécialisation : la première comporte la fonction de multiplication, tandis que la seconde gère les instructions multimédias inédites avec le *21264* (elles administrent l'encodage MPEG-2 en temps réel). Il faut savoir qu'un processeur est amené à effectuer d'autres calculs que ceux qu'un programme spécifie. Il doit en effet calculer les adresses de la mémoire à laquelle il lui faut accéder. Voilà pourquoi les unités de chargement se révèlent aptes à faire des additions. Les unités entières sont affectées à deux groupes, comprenant chacun une unité d'accès à la mémoire et une autre de calcul général, ainsi qu'un ensemble de quatre-vingts registres. Les deux ensembles de registres se voient synchronisés entre eux, puisque les instructions manipulant les véritables registres peuvent être affectées à l'un ou l'autre des groupes.



*L'Alpha.*

Les caches instructions et données comprennent maintenant 64 Ko chacun, tandis que le cache de niveau 2 disparaît. Vu la fréquence du processeur, l'accès au cache s'opère en deux cycles. C'est l'un des premiers processeurs à opter pour ce choix. Sur le *21164*, le processeur pouvait accéder au cache en un seul cycle, mais la petite taille des caches faisait qu'il échouait assez souvent (la donnée ou l'instruction recherchée ne se trouvaient pas dans le cache). Il lui fallait donc accéder au cache de niveau 2, ce qui consommait six cycles, sans compter que cette démarche imposait de fréquents échanges entre les deux niveaux de cache. Une technique de préchargement des caches depuis la mémoire compense largement cette contrainte. D'autre part, il y a moyen d'anticiper jusqu'à trente-deux lectures et trente-deux écritures, tandis que quatre-vingts instructions peuvent

**Les technologies de fabrication**

***Chaque génération de microprocesseurs est définie par son processus de fabrication. Celui-ci détermine la densité des transistors en fonction du pas de grille qu'il met en œuvre et du nombre de niveaux de métal. Plus le pas de la grille est petit, plus on pourra disposer de transistors sur une surface donnée (de l'ordre de 2 à 3 cm² pour la majorité des processeurs), ce qui permet d'augmenter la complexité du processeur. De même, les liaisons entre ces transistors se raccourciront, ce qui se traduira par des fréquences plus élevées. Ainsi, au début des années 1990, le pas de la grille était de 1 micron avec 2 niveaux de métal ; ce résultat donnait lieu à des fréquences inférieures à 100 MHz, pour un capital d'environ 1 million de transistors. La technologie actuelle permet de fabriquer des microprocesseurs comprenant un pas de grille de 0,25 microns et jusqu'à six niveaux de métal, soit jusqu'à 20 millions de transistors, lesquels engendrent des fréquences de 500 à 600 MHz. La prochaine technologie appliquée vers la fin de l'année accédera au gigahertz et comportera 100 millions de transistors, ainsi qu'un pas de grille de 0,18 microns. Le métal employé aujourd'hui est de l'aluminium, mais IBM a commencé à produire en série des processeurs utilisant du cuivre (il en va ainsi du PowerPC G3 à 400 MHz). Ce métal offre une conductivité accrue, ce qui devrait favoriser encore l'amélioration des performances, bien qu'il soit plus complexe à employer dans le cadre de la microélectronique.***

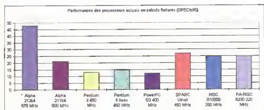



*Tableau comparatif des calculs en flottants.*

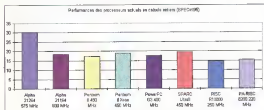



*Tableau comparatif des calculs en entiers.*

être en cours de traitement.

Un problème que rencontre tous les microprocesseurs aujourd'hui vient de la relative lenteur de la mémoire. Par conséquent, les processeurs passent de plus en plus de temps à attendre la mémoire. Le *21264* requiert environ quatre-vingts cycles, pour accéder à la mémoire. Au-delà des techniques de préchargement, un bus de 64 bits à 333 MHz est utilisé pour accéder à la mémoire. Celui-ci est démultiplexé sur 256 bits, afin qu'il devienne capable d'exploiter de la mémoire standard à 100 MHz. Ce procédé permet d'atteindre une bande passante de 2 Go/s. L'accès à un cache de niveau 2 s'effectuera si besoin à 333 MHz, ce qui autorise la présence d'une bande passante de 5 Go/s.

Pour réduire encore les temps d'attente liés à la mémoire, l'utilisateur a loisir d'exécuter les instructions en ne respectant pas l'ordre établi par le programme. Cette situation permet d'anticiper les accès mémoire et donc de disposer des données dans les registres avant d'en avoir besoin.

Le *21264* prédit les branchements avant qu'ils ne se produisent. Ce mécanisme existait déjà dans le *21164*, mais il a été grandement amélioré. Les branchements dans un programme diminuent les performances des processeurs, car ils obligent à vider les pipelines. Il est donc essentiel d'essayer de les anticiper. Pour ce faire, le processeur doit pouvoir prédire quelle sera la prochaine instruction à exécuter.

**Le futur**

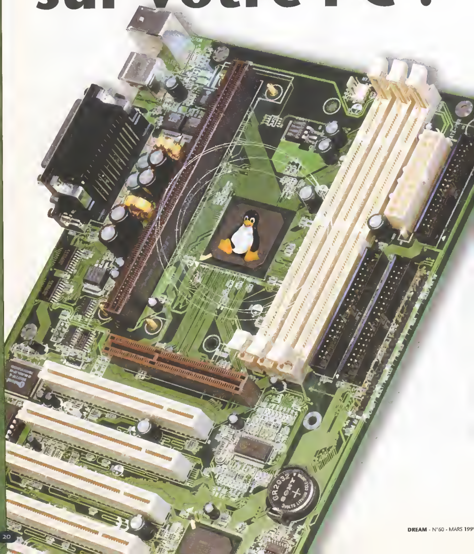
Le successeur du *21264* a déjà été présenté lors du Microprocessor Forum, les 13 et 14 octobre 1998. Le *21364* (*EV7*) sera construit autour du même cœur que le *21264*, mais avec des fréquences plus élevées, puisque la première version entend afficher 1,1 GHz. La réduction du pas de la grille à 0,18 microns donnera à l'utilisateur le privilège de disposer d'un potentiel de transistors beaucoup plus important, puisqu'il devrait atteindre les cent millions, dont quatre-vingt-douze millions de transistors pour la mémoire intégrée. Ce potentiel sera exploité pour intégrer plusieurs composants de base, tel qu'un cache de niveau 2 de 1,5 Mo, ainsi qu'un contrôleur mémoire et un contrôleur réseau. L'intégration du cache de niveau 2 permettra de posséder une bande passante processeur/cache de l'ordre de 20 Go/s au lieu des 5 Go/s obtenus par le *21264* avec un cache externe. Le contrôleur mémoire qui fonctionnera à 800 MHz conférera un débit de 13 Go/s au lieu des 2 Go/s actuels. On exploitera le contrôleur réseau pour connecter plusieurs processeurs entre eux avec une bande passante de 10 Go/s ; celui-ci servira également d'interface vers le bus d'entrées/sorties, avec une bande passante de 3 Go/s. Les estimations de performances pour ce processeur tournent autour de 85 SPECint95 et 130 SPECfp95. Il devrait être échantillonné à la fin de cette année et produit en volume vers la fin de l'an 2000. La vraie révolution aura lieu un peu plus tard, probablement à l'horizon 2001/2002, avec l'*EV8* ou *21464*. Ce nouveau produit sera multithread, ce qui signifie que le niveau d'intégration permis par la technologie (pas de grille de 0,15 microns, environ 250 millions de transistors) autorisera la mise en œuvre de plusieurs processeurs à l'intérieur d'une seule puce. La première implémentation devrait gérer quatre threads simultanés ; cela veut dire que le processeur comprendra quatre compteurs de programme indépendants. Sa fréquence devrait tourner autour de 1,4 à 1,5 GHz. Atteindre les 200 SPECint95 et dépasser les 300 SPECfp95 constituent les principaux objectifs. Le processeur Merced d'Intel et son successeur le McKinley devraient péniblement parvenir au stade des 100 SPECint95, à la même époque.

*Karine Mordal*

**Les principaux microprocesseurs actuels**

| Nom | Fréquence (MHz) | SPECint95 | SPECfp95 | Cache (i/d) (Ko) | Technologie (micron) | Transistors (millions) |
|---|---|---|---|---|---|---|
| Alpha 21264 | 575 | 30,3 | 47,7 | 64/64 | 0,35 | 15,2 |
| Alpha 21164 | 600 | 18,4 | 21,3 | 8/8/96 L2 | 0,35 | 9,3 |
| Pentium II | 450 | 17,2 | 12,9 | 16/16 | 0,25 | 7,5 |
| Pentium II Xeon | 450 | 18,9 | 14,7 | 16/16 | 0,25 | 7,5 |
| PowerPC G3 | 400 | 17,6 | 12,2 | 32/32 | 0,22 | 6,35 |
| UltraSparc II | 450 | 19,6 | 27,1 | 16/16 | 0,25 | 5,4 |
| PA-RISC 8200 | 220 | 15,5 | 25 | | 0,5 | 3,8 |
| MIPS R10000 | 250 | 14,7 | 24,5 | 32/32 | 0,35 | 6,8 |

# Installez la Red sur votre PC !



# Hat Linux 5.2

*Dream inaugure sa nouvelle formule en fanfare nous vous offrons ce mois-ci la fameuse distribution Red Hat 5.2 de Linux sur notre CD-Rom. Voici comment l'installer.*

Linux, système d'exploitation totalement gratuit, s'affirme désormais comme une réelle alternative à *Windows*. Après être resté quelques temps un système réservé aux connaisseurs, il devient à travers ses différentes distributions et ses nombreuses interfaces graphiques, comme Kde par exemple, un système facile à installer et simple à utiliser. Le succès et la puissance de *Linux* attirent d'ailleurs de plus en plus de personnes, simples particuliers comme entreprises. Certains éditeurs du monde *Windows* franchissent même le pas, à l'image de Corel, qui met à disposition *WordPerfect 8* pour *Linux*, de même que d'autres logiciels puissants et gratuits, tels que Gimp, le *Photoshop* de *Linux*, et *StarOffice 5*, la suite bureautique capable de remplacer totalement *Office 97*, les bugs en moins.

**Faire cohabiter Linux et Windows**

*Linux* peut devenir dès aujourd'hui votre nouveau système d'exploitation ou bien encore s'installer en plus de votre OS actuel (a priori, *Windows*) sans problème. Il vous faudra simplement respecter une règle : *Linux* ne s'installe pas dans un fichier *Windows*. C'est un autre système d'exploitation ; il demande donc sa propre partition. Ainsi, si vous décidez d'installer *Linux*, votre première étape consistera à lui trouver une partition personnelle.

Tout dépend ici de la configuration de votre ordinateur. Si vous possédez deux disques durs, le

### Une installation automatique pour l'utilisateur pressé

*Les connaisseurs remarqueront sans doute la nouveauté introduite dans le choix de l'installation. Vous pouvez désormais opter pour l'une de ces trois classes d'installation : station de travail, serveur ou personnalisée (installation classique). La différence se situe par rapport aux packages. Dans les deux premiers cas, vous n'aurez pas l'opportunité de procéder à votre sélection. L'autre différence vient des partitions qui seront créées sur le disque, soit trois pour un total de 600 Mo pour la première, contre six sur 1,6 Go pour la seconde.*

*Attention, une installation "station de travail" détruira toutes les partitions Linux existantes sur votre disque et un "serveur" réorganisera la totalité de l'ordinateur : elle supprimera toutes les partitions sans distinction de type. Adieu votre Windows... Cependant, ces classes se trouveront en mesure de vous rendre le travail moins fastidieux. L'utilisateur pressé optera lui aussi avec ravissement pour ces méthodes qui accélèrent nettement l'installation.*



plus judicieux serait de consacrer le premier disque à *Windows* et le second à *Linux*. Pour ceux qui n'ont qu'un seul disque dur, peut-être possédez-vous déjà plusieurs partitions. Pour le savoir, il n'y a rien de plus simple ; regardez sous *Windows* le nombre de lecteurs que vous possédez. Vu que le lecteur C correspond à celui réservé à *Windows*, essayez de libérer (vider totalement) un des autres lecteurs. Si vous n'avez qu'un lecteur, il va alors vous falloir repartitionner votre disque. Attention, si vous utilisez *Fdisk*, les données de votre disque dur seront entièrement perdues. Exploitez plutôt un des utilitaires disponibles, comme *Partition Magic*, qui vous aideront à créer une nouvelle partition sans pour autant tout perdre. N'oubliez pas que les données de la partition sur laquelle vous installerez *Linux* seront effacées ; aussi, profitez de cette installation pour faire un peu de ménage. Le but est de libérer environ 1 Go pour *Linux*.

**Votre matériel est-il compatible ?**

Avant de commencer l'installation tête baissée, vérifiez la compatibilité de votre matériel avec *Linux*, notamment en ce qui concerne la carte vidéo. Pour cela, vous pouvez consulter le site Web de Red Hat (http ://www redhat.com), mais à moins de posséder la toute dernière carte diffusée sur le marché, votre matériel devrait être reconnu. En cas de problème avec le driver de la carte vidéo, vous avez toujours moyen de faire un tour sur le Web (http://www xfree86.org) pour y trouver le dernier *XFree86*, la version libre du serveur *XWindow*, qui gérera peut-être votre carte la 3.3.2.3 correspond en effet à la version de la Red Hat fournie sur le CD-Rom ou bien encore de récupérer le patch, présent sur le CD du numéro précédent.

Quoi qu'il en soit, préparez la liste de votre matériel avec la carte son, la carte vidéo, la fréquence de balayage de votre écran, éventuellement la carte SCSI et le port sur lequel est branché votre modem. Il vous manque des informations ? Vous pouvez les trouver dans le panneau de configuration de *Windows* (vous savez, l'icône système...).

**Qu'est-ce que "Red Hat" ?**

Parmi toutes les distributions de *Linux*, la version Red Hat semble être devenue à la fois la plus classique et la plus populaire. Le programme d'installation se révèle quant à lui très simple d'utilisation et parle français, de surcroît ! ! ! On peut décrire son fonctionnement de la manière suivante : la touche *Entrée* sert à valider un écran, la touche de tabulation, à se déplacer dans les différents champs de l'écran, la barre d'espace, à sélectionner/désélectionner et les flèches, à se déplacer dans une liste. Ainsi, au bout du deuxième écran, on se sent déjà à l'aise. Prenez garde toutefois à ne pas confondre vitesse et précipitation : si vous appuyez plusieurs fois de suite sur la touche *Entrée*, vous risquez de valider les écrans postérieurs sans avoir été en mesure de choisir quoi que ce soit à l'intérieur ! ! ! La gestion des fichiers .RPM qui a fait beaucoup pour la notoriété de cette distribution est bien évidemment toujours présente (voir encadré).

Dernière petite précision : avant de vous lancer dans l'aventure *Linux*, il convient de préciser que cette version repose sur le noyau 2.0.36 de *Linux* mais vous pourrez ensuite installer le nouveau noyau de *Linux*, une fois que vous vous serez familiarisé avec votre nouveau système d'exploitation.

*Karine Morda*

### Les RPM

*La Red Hat utilise un système simple pour faciliter les installations : les RPM, c'est-à-dire des packages contenant un ensemble de fichiers. Pour obtenir la liste complète des options de la commande rpm, il vous suffit de tapez rpm --help. Prenons un exemple. Imaginons que vous vouliez installer le logiciel Toto qui se trouve sous la forme Toto.rpm. Tapez :*

```
rpm -ivh toto.rpm
```

ou bien s'il s'agit d'une mise à jour

```
rpm -uvn toto.rpm
```

Pour le désinstaller, tapez :

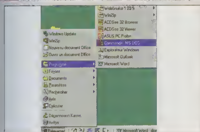
```
Rpm -e toto
```

### Lilo

*Lilo désigne le chargeur de boot le plus couramment utilisé. Prenez garde, pourtant : si vous installez Linux au-delà du 1023ème cylindre de votre disque, Lilo risque de ne pas pouvoir activer Linux. Une astuce consiste à créer une partition Linux avant le cylindre 1023. D'une taille de 10 Mo, elle aura pour point de montage /boot et contiendra le noyau et les fichiers de démarrage. Le reste du système pourra alors être installé après ce cylindre fatidique.*

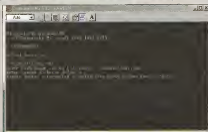
### Partitionnez en connaissance de cause

*Il vous faudra créer au minimum une partition Linux native et une partition swap. Le swap est un espace du disque, utilisé comme mémoire auxiliaire pour stocker les données ou les programmes inactifs. En général, la taille de la swap est définie par rapport à celle de la Rom : la première représente le double de la seconde, avec un maximum possible de 127 Mo. Quant à la partition Linux native, il peut se révéler plus astucieux d'en créer plusieurs : une pour la racine qui contiendra le système, une seconde pour le home abritant tous vos fichiers personnels et une dernière pour le /usr/local recelant les logiciels. Lorsque vous effectuerez une mise à jour du système, vous n'aurez pas à craindre la disparition de vos données personnelles, et les logiciels anciennement installés continueront à fonctionner. Les partitions sont baptisées "hda1", puis "hda2", etc. pour le premier disque, le CD-Rom prend le nom de "hdb" et le second disque s'appellera "hdc", en observant le même principe de partitions. Si vous souhaitez élaborer plus de quatre partitions, il vous faudra en créer une plus étendue que les autres, avant de la découper ensuite en lecteurs logiques dont les noms commenceront à partir de hda5. En ce qui concerne les disques SCSI, ils se nommeront "sda", "sdb", etc, selon le même principe.*



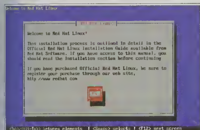

**On part de Windows...**

La méthode infaillible pour démarrer une installation de *Linux* consiste à créer une disquette de démarrage Ouvrez une fenêtre *DOS*, placez-vous dans le répertoire d'outils du CD et tapez la commande "rawrite".



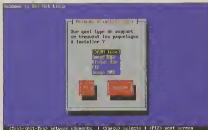
**Création de la disquette de boot**

Au moment où l'ordinateur vous demande d'entrer le nom de l'image source pour la disquette, tapez : "..\images\boot.img", puis "A" pour le lecteur de disquette. Insérez la disquette et tapez sur la touche *Entrée*. Redémarrez.



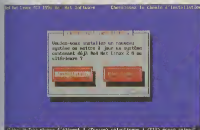

**On commence !**

Une fois que l'ordinateur a booté sur la disquette, un message apparaît, qui vous demande en quel mode vous voulez lancer l'installation ; appuyez sur la touche *Entrée* et laissez-le continuer. Cet écran apparaîtra alors.



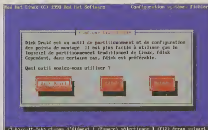

**Quelques précisions supplémentaires**

Vous choisirez votre langue, le type de clavier utilisé (fr-latin1) et le support sur lequel se trouve la Red Hat : CD-Rom local. Attention, votre lecteur de CD doit être branché en IDE ou en SCSI, mais pas sur la carte son.



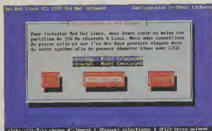

**Mode d'installation**

Ici, un seul choix : installation. A l'écran suivant, choisissez personnalisée comme type d'installation.




**Disk Druid ou Fdisk ?**

C'est à ce niveau que se pose réellement la question des partitions pour *Linux*. Un seul choix possible pour être sûr du résultat s'impose : *Fdisk*.

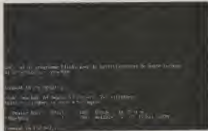






**Premiers pas avec Fdisk**

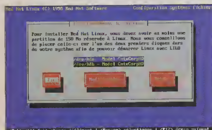
Choisissez le disque dur contenant une partition de 1 Go réservée à *Linux*. Tapez "p" pour afficher les partitions. Si vous avez un disque entier à consacrer à *Linux*, effacez-les toutes en tapant "d", puis le numéro de la partition...






**Création des partitions**

Vous disposez d'une première partition, de type *DOS* ou *Windows* et voulez installer *Linux* à la suite. Dans ce cas, ne gardez que la première partition et effacez les autres ("d" et le numéro). Pour quitter *Fdisk* sans rien sauvegarder, tapez "q"

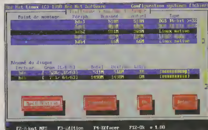






**On continue...**

Si vous avez à redémarrer votre machine pour que les changements deviennent effectifs, recommencez la même procédure d'installation jusqu'à l'étape cinq, choisissez *Fdisk* et revenez ici. A présent, appuyez sur "Fin"

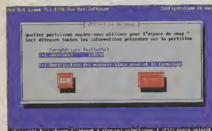






**On avance...**

Sélectionnez la première partition *Linux* native et appuyez sur modification.

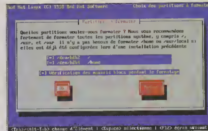






**Activation de la swap**

Sélectionnez les deux options (il faut que l'astérisque apparaisse devant chacune d'elles) et appuyez sur "Ok". La partition swap se formate alors pendant un court instant.







**Formatage des partitions**

Sélectionnez ici aussi toutes les options pour effectuer un formatage complet et efficace. Cliquez sur "Ok".



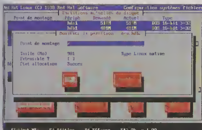
### 8 Une première partition

Nous allons créer trois partitions. Pour en ajouter une, tapez "n", "p", le numéro de la partition à créer, le premier cylindre proposé et enfin la taille souhaitée, ("+900M"). Le principe reste le même pour les suivantes.



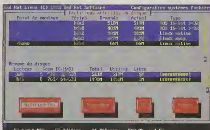
### 9 La swap

Il nous reste maintenant à changer le type de la seconde partition, en tapant simplement "t" puis 82. Une dernière vérification s'impose : tapez "p" et si vous avez un écran qui ressemble à celui-ci, tapez w pour enregistrer ces changements.



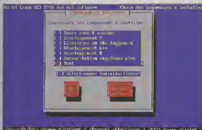

### 12 Point de montage

Ici, on précise simplement comment sera montée cette partition. Inscrivez "/" dans la première cellule de texte, puis validez avec "Ok". Recommencez la même opération avec l'autre partition *Linux* native et, cette fois-ci, tapez "/home".



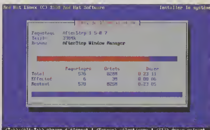

### 13 Fin du partitionnement

Voilà le résultat final ! Appuyez sur "Ok" si vous parvenez à la même chose ; autrement, il vous reste l'option "Retour" pour revenir en arrière (étape 5 et recommencez).



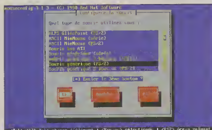

### 16 Choix des composants

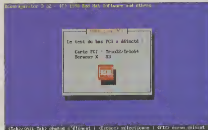
La méthode la plus simple consiste à faire défiler la liste jusqu'au dernier élément. Vous cochez la case "tout" et validez. Vous pourrez toujours par la suite désinstaller certains composants, grâce à la gestion de .RPM de la Red Hat.




### 17 Ça s'installe...

L'installation devrait se dérouler sans problème. Toutefois, vous recevrez tout de même trois messages au sujet de l'installation de *Netscape Communicator*. Il suffit juste de valider les messages et l'installation continuera normalement.



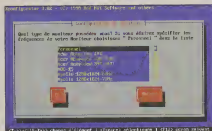



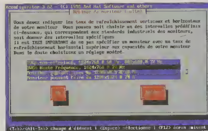

**La souris**

18 Après avoir validé le message de détection de la souris, vous arrivez à cet écran. A ce moment précis, vérifiez qu'il s'agit de la bonne souris ; si celle-ci ne possède que deux boutons, sélectionnez l'option "Emuler le troisième bouton".

**La carte vidéo**

19 Beaucoup d'entre vous auront la chance de voir s'afficher un écran similaire. Pour les autres, la démarche devient un petit peu plus délicate : la Red Hat vous proposera une liste de cartes dans laquelle vous retrouverez la vôtre.



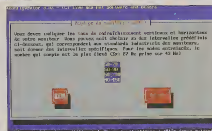



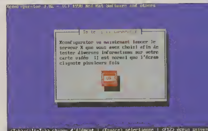

**Choix de l'écran**

20 Cherchez votre écran dans la liste. Cependant, si vous ne connaissez pas son type exact ou s'il n'apparaît pas dans la liste, cela n'a pas une grande importance. Choisissez "personnel" et continuez.

**Taux de rafraîchissement**

21 Indiquez le taux de rafraîchissement de votre écran. En cas de doutes, choisissez la même définition que celle utilisée par *Windows*. Une fréquence trop élevée peut endommager votre matériel ; aussi, ne rentrez pas n'importe quelle valeur !



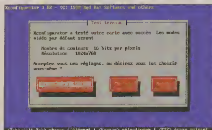




**Dernière étape pour l'écran**

22 Vous devez préciser l'intervalle de fréquences que votre écran peut supporter. Pour les écrans ordinaires, l'intervalle 50-70 est le plus courant.

**Recherche du chipset, de l'horloge...**

23 *Xconfigurotor* va tester le serveur X que vous avez auparavant installé (avec le choix de la carte vidéo). Pour les moins chanceux, le test échouera mais il suffira alors de rentrer les valeurs manuellement.







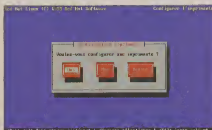
**Test réussi**
Si vous arrivez à cet écran, choisissez ces valeurs par défaut.







**Configuration du réseau**
Répondez non à cette question, sauf si vous êtes effectivement en réseau. Attention, il ne s'agit pas ici de la configuration du modem.







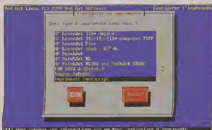
**L'imprimante**
Les étapes suivantes ne concernent que ceux qui veulent configurer leur imprimante sous *Linux*. Pour les autres, passez directement à l'étape 35.







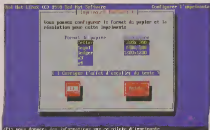
**Où se trouve l'imprimante ?**
Choisissez "port local", à moins que votre imprimante ne se trouve pas sur le port parallèle de votre PC.







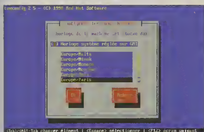
**Nom de l'imprimante**
Sélectionnez votre imprimante dans la liste ou "imprimante PostScript" si elle n'existe pas.



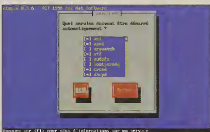




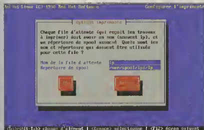
**Format d'impression**
Choisissez les réglages qui correspondent à votre imprimante. Par la suite, vous pourrez également reparamétrer correctement votre imprimante, au cas où vous ne seriez par sûr des bonnes valeurs.



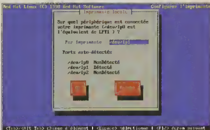

**26** **Quelle heure est-il ?**
Choisissez "Europe Paris" pour la France.



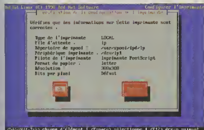

**27** **Modules lancés au démarrage**
Laissez les modules qui sont sélectionnés par défaut ; vous pourrez toujours changer d'avis plus tard, s'il vous prend l'envie de recompiler le noyau, par exemple.



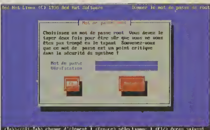

**30** **Chemin d'accès**
Laissez les paramètres affichés par défaut.



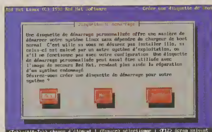

**31** **Le port de l'imprimante**
Laissez également les valeurs indiquées. En général, il s'agit des bonnes valeurs et vous n'avez rien à modifier.




**34** **L'imprimante est paramétrée**
Voici un joli résumé des valeurs de votre imprimante.

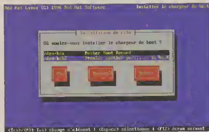



**35** **Le mot de passe**
Tapez le mot de passe root de votre choix sur six caractères, une première fois puis une seconde sur la ligne suivante. Vous pourrez en changer plus tard ou ne pas en introduire du tout, mais pour l'instant, il vous est indispensable.



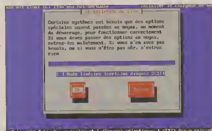

**36** **La disquette**

Créez une disquette de démarrage : elle peut vous rendre service si vous ne réussissez pas à installer *Lilo* (*Lilo* désigne le programme qui vous permet de choisir entre *Linux* ou *Windows* au moment d'activer votre ordinateur).



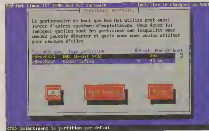

**37** **Lilo**

*Lilo* a besoin de s'installer sur la première partition du premier disque dur "Master boot record" (voir encadré).



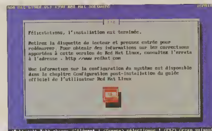

**38** **Options de Lilo**

Par défaut, ne placez rien sur cette ligne.




**39** **Partitions de boot**

Ces noms correspondent à ce qu'il vous faudra taper lors du démarrage de *Lilo* ; il s'agira de *DOS*, si vous voulez lancer *Windows*, et de *Linux*, au cas où vous souhaiteriez explorer votre tout nouveau système




**40** **Félicitations !**

Que dire de plus ? Vous avez finalement atteint le bout du tunnel : ce n'était pas si difficile !



**41** **On redémarre...**

Une fois votre PC redémarré, votre écran vous affichera *LILO Boot*. A vous de choisir. .

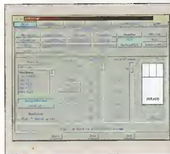
# Configurez X11 en quelques clics de souris

*Maintenant que vous avez installé Linux et que vous avez fait le tour du propriétaire en mode console, vous aimeriez sûrement passer en mode graphique.*

Dans le monde de *Linux*, le mode graphique se prononce *X-Window* (sans le S) ou bien encore *XFREE86*. Ce dernier correspond à la version libre du serveur *X11* implémenté pour *Unix*. Par-dessus cette première couche, en général, chacun ajoute son *Window* Manager fétiche. Cette architecture particulière a plusieurs conséquences. La plus visible est la séparation qui existe entre le matériel et le logiciel. Ainsi, une fois votre serveur *XFREE* configuré, n'importe quel *Window* Manager fonctionnera sans avoir à lui indiquer toute la configuration matériel de la machine. Chaque utilisateur peut ainsi employer *Afterstep Window Maker* ou encore *Kde*, par exemple, sans avoir à se soucier du *Window* Manager de son voisin. Vous pourrez ainsi vous créer plusieurs comptes utilisateurs et employer un *Window* Manager différent pour chacun d'entre eux. Mais laissons de côté les considérations théoriques et autres architectures client/serveur du monde graphique de *Linux* pour nous pencher sur votre principal sujet de préoccupation : comment configurer *Xfree* pour pouvoir disposer enfin d'un environnement graphique agréable.

Ceux qui ont installé la Red Hat ont déjà fait connaissance avec l'utilitaire de cette distribution : *XConfigurator*. Mais pour ceux qui ne sont pas satisfaits des premiers réglages ou dont le programme n'a pas détecté la carte, il reste une autre possibilité : *XF86Setup*. Une fois logué en mode root, tapez simplement *XF86Setup*. Respectez bien entendu la casse (majuscule/minuscule) si vous voulez voir apparaître quelque chose. Après quelques questions ("voulez-vous charger la configuration existante ?" à laquelle il vaudrait mieux répondre "non" pour éviter tout problème) et un petit temps d'attente au cours duquel le serveur de base SVGA est lancé, vous verrez apparaître un utilitaire graphique (eh oui !) dans lequel vous n'aurez plus qu'à cliquer pour configurer votre machine.

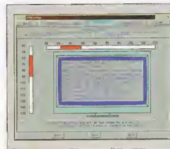
*Karine Mordal*





**La souris**

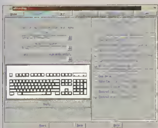
Cliquer sur l'onglet *Mouse*. Une première fenêtre s'ouvre ; vous pouvez la fermer en cliquant sur *Dismiss*. En dessous de cette fenêtre se situe la fenêtre de paramétrage de la souris. Les paramètres du port de connexion souris (/dev/psaux) et du type de souris (ici PS/2) sont rarement à changer. N'oubliez pas d'émuler votre souris à trois boutons si elle n'en possède que deux. Au fur et à mesure que vous utiliserez un environnement graphique, vous en découvrirez tous les avantages. Faites un essai avec les boutons de la souris pour vérifier la configuration de celle-ci.

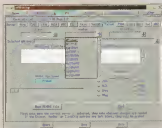




**L'écran**

Choisissez dans la liste centrale le mode qui correspond à votre écran. Le plus courant pour les écrans de moyenne gamme est Non interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz. Faites attention : lisez la documentation de votre écran avant de rentrer les valeurs. Un mauvais choix pourrait endommager votre écran. Dans le doute, commencez par des valeurs basses (Super VGA 800x600 Hz notamment) avant d'en adopter de plus élevées. Vous pouvez également affiner vous-même les paramètres "horizontal" et "vertical", en diminuant par exemple la longueur de l'intervalle. Cliquez ensuite sur l'onglet *Modeselection*.





**2** **Le clavier**

Si la souris est bien configurée, cliquez sur *Apply* puis sur l'onglet *Keyboard*. Ici, il n'y a pas grand-chose à faire. Dans *modèle*, sélectionnez *Generic 101-key Pc* ou *102, 103, 104, 105* ou bien encore *natural Keyboard* selon votre clavier. Pour obtenir la liste, il suffit de cliquer sur la flèche. Dans *Layout*, choisissez *French*. Laissez les différentes options qui se trouvent dans les fenêtres de droite telles quelles et cliquez sur le bouton *Apply*, puis sur l'onglet *Card*.

**3** **La carte**

Cliquez sur le bouton *Card List* pour sélectionner votre carte vidéo parmi la liste détaillée. Une fois votre carte sélectionnée, cliquez sur *Detailed Setup* pour la configurer. Le serveur a été choisi en fonction de votre carte. Laissez-le. Il vaut mieux choisir l'option "Probed" dans les cases *Chipset*, *RamDac* et *ClockChip*, à moins que vous ne possédiez la documentation de votre carte et que vous soyez sûr des valeurs à donner. Sélectionnez la quantité mémoire vidéo que vous possédez. Cliquez maintenant sur l'onglet *Monitor*.





**5** **Le nombre de couleurs**

Choisissez ici la résolution de votre écran. Vous pouvez sélectionner plusieurs résolutions (ici, il y en a trois). Au cours du chargement de X, la meilleure résolution sera lancée et vous pourrez également basculer grâce à la combinaison de touches Ctrl-Alt-+. Sélectionnez ensuite le nombre de couleurs affichées : 8bpp, 16 Bpp, etc., du plus petit nombre au plus grand. Une fois votre choix effectué, cliquez sur *Done*. Un test est effectué pour savoir si le serveur choisi s'avère correct. Si tel est le cas, vous arriverez à l'écran suivant. Autrement, recommencez en vérifiant un par un vos paramètres.

**6** **C'est fini**

Voilà, ca y est. vous allez enfin pouvoir disposer d'un environnement agréable. Il ne vous reste plus que quelques réglages à effectuer, comme le centrage de votre écran, en utilisant *xvidtune*. Cependant, prenez garde : n'essayez pas n'importe quoi et soyez prudent avec cet utilitaire. Vous n'avez plus ensuite qu'à cliquer sur "Save the configuration and exit" pour enregistrer les paramètres de votre configuration dans /etc/X11/XF86Config. Ensuite, tapez "startx" pour lancer X et voir le résultat de vos réglages. Vous pouvez également relancer *XF86Setup* si la configuration a échoué ou affiner vos réglages avec la commande xvidtune.

# Kde, installation et présentation d'un environnement graphique complet

*Et maintenant, passons à la dernière étape, pour disposer enfin d'un vrai environnement graphique, muni de tout son confort, et pouvoir prétendre que Linux a des chances de devenir un système intuitif... ou presque.*

Avant de se trouver en mesure de cliquer partout, il reste une dernière petite corvée : l'installation de *Kde*. Après avoir monté le CD dans le répertoire /mnt/cdrom par exemple, placez-vous dans le répertoire qui contient *Kde* et installez-le :

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
cd /mnt/cdrom/Linux/Kde-qt/
rpm -i qt*
```

Tapez ensuite la même commande rpm pour les fichiers kde, en respectant tout de même un certain ordre. Les fichiers kdelibs, kdesupport et kdebase doivent être installés avant les autres. Tapez ensuite ces quelques lignes pour faire de *Kde* votre *Window* Manager :

```
cp /opt/kde/bin/startkde /home/vous/.xinitrc
chmod a+x .xinitrc
```

Vous avez ainsi recopié le script de démarrage de *Kde* dans votre script de démarrage graphique personnel.



*Kfm, version icônes à la Windows.*

Editez ensuite le fichier .xinitrc et insérez cette ligne au début de celui-ci :

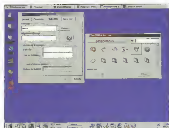
```
export PATH=$PATH:/opt/kde/bin
```

Cette commande ajoute dans votre PATH un répertoire supplémentaire, qui correspond à l'endroit où se situent les commandes de *Kde*. Tapez à présent la commande startx pour découvrir enfin *Kde*.

**La barre de menus**

Enfin ! Vous y êtes. Voici un environnement qui ressemble à ce que les utilisateurs de *Windows* connaissent. Pour preuve, cliquez donc sur le K. Pour changer le menu qui apparaît, il vous suffira de cliquer sur "panneau". Voilà une petite astuce supplémentaire pour les utilisateurs de Red Hat. En mode root, tapez :

```
wmconfig --output kde
mv /root/.kde/share/applnk/redhat /opt/kde/share/applnk
```



*Ajout d'une nouvelle icône.*

Relancez *Kde* et vous verrez alors apparaître une nouvelle ligne dans le menu. Pratique, non ? Toutefois, vous aimeriez sans doute configurer votre bureau de manière plus personnelle... Pour cela, une seule chose à retenir : le centre de contrôle.

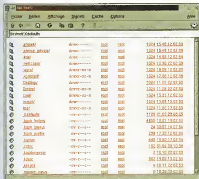
**Panneau de configuration ? Centre de contrôle !**

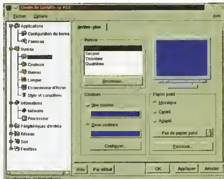
Le centre de contrôle regroupe en une seule fois les différents utilitaires situés dans le menu configuration. Ici, tout se révèle intuitif : un clic de souris et c'est réglé. Vous pouvez configurer votre bureau, en commençant par le fond d'écran, l'économiseur ou encore les sons. A moins que vous ne vouliez changer le style des fenêtres. A vous de jouer.

**Se loguer en mode graphique**

*Kdm* est un Display Manager (une fenêtre pour se loguer et arriver directement dans un environnement graphique) plus agréable que le classique *Xdm*. Pour configurer *Kdm*, allez dans "configuration" application et "configuration du bureau". Il vous faudra ensuite avertir le système pour qu'il lance *Kdm* au démarrage. Dans cette perspective, vous devrez effectuer quelques petites manipulations supplémentaires. La première chose à vérifier est l'existence du fichier .xsession dans son home. S'il fait défaut, créez-le et glissez-y cette simple ligne : exec .xinitrc. Ensuite, éditez le fichier /etc/inittab et opérez ces deux petits remplacements :

```
remplacez id:3:initdefault:
par id:5:initdefault:
puis remplacez x:5:/usr/bin/X11/xdm -nodaemon
par x:5:/opt/kde/bin/kdm -nodaemon
```


Kfm version détails et listes à la Linux.


Le centre de contrôle.

Relancez le système et le petit pingouin de *Linux* devrait apparaître.

**Des icônes comme pour Windows**

Bon, d'accord, mais comment fait-on pour rajouter des icônes sur le bureau ? Rien de plus simple ! Un clic droit sur le bureau, on sélectionne "nouveau" et "programme", on indique le nom que l'on souhaite lui donner et un nouvel icône apparaît. Cliquez dessus avec le bouton droit pour le configurer : informez le programme auquel il correspond, changez le dessin de l'icône si vous le souhaitez et vous avez terminé ! Pour les périphériques, c'est la même chose. Effectuez un clic droit, puis choisissez "nouveau", et "device" Ensuite, un second clic, "propriété" et il ne vous reste plus qu'à renseigner le périphérique voulu. Pour le CD-Rom par exemple, tapez "/dev/cdrom" dans le périphérique. Sans doute aurez-vous encore d'autres détails à configurer selon vos envies, mais vous vous rendrez compte qu'au fond, *Kde* est très convivial, même et surtout lorsqu'on débute et que l'on veut découvrir *Linux*.

**Kfm, le gestionnaire à tout faire**

Continuons la visite de *Kde* avec *Kfm*, le gestionnaire de fichiers très pratique. Afin d'en avoir un premier aperçu, cliquez sur l'icône avec une petite maison située dans la barre de menus. Une fenêtre s'ouvre avec à l'intérieur vos fichiers et répertoires. A vous de configurer l'affichage selon vos préférences : à la *Windows* avec les icônes ou à la *Linux* pur et dur, avec le détail des fichiers. Pour parcourir vos fichiers, un simple clic suffit, que ce soit pour afficher le contenu d'un répertoire, un fichier texte ouvert grâce à l'éditeur de texte ou bien encore tout autre fichier dont l'extension correspond à un lien que vous aurez défini (voir encadré). Cependant, *Kfm* ne s'arrête pas là. Il vous servira aussi de browser Web, d'afficheur de pages Html... Bref, il deviendra très vite un outil indispensable, même s'il peut dérouter au début les habitués de *Linux*. On a ainsi moyen de faire du Ftp tranquillement, en tapant l'URL de son choix (ftp://ftp.somehost.com.), puis en procédant à un copier-coller, de cette fenêtre vers une autre locale. Si vous désirez vous connecter en vous authentifiant, entrez alors : ftp://username@ftp.somehost.com. *Kfm* vous demandera alors votre mot de passe et vous connectera au serveur.

**A chacun son Kde**

Pour les puristes qui ne supportent pas les clics de souris et qui veulent retrouver leur bon vieil *Xterm*, *Kde* a pensé à eux en leur offrant un terminal amélioré. Pour le découvrir, cliquez sur l'icône représentant un terminal dans la barre de menus. De toute façon, quelle que soit votre façon de travailler, *Kde* est suffisamment configurable pour plaire à un grand nombre d'utilisateurs. Le seul petit reproche que l'on pourrait lui faire vient d'un substantiel besoin de ressources. Mais après tout, un environnement graphique demande souvent beaucoup de ressources, surtout lorsqu'on lui demande de pouvoir tout contrôler par un simple clic de souris. Alors, à chacun son aventure et à chacun sa découverte de *Linux* ; pensez à nous envoyer des captures d'écran de vos bureaux les plus sympathiques.

*Karine Mordal*

## Les MIME de Kde

*Pour associer un type d'extension de fichiers à une application, lancez Kfm, puis cliquez sur "édition", "type mime". Vous arrivez alors dans le répertoire qui contient vos liens personnels. Créez un nouveau type Mime ("fichier", "nouveau", "type Mime"), cliquez avec le bouton de droite et sélectionnez "propriétés". Pour remplir toutes les lignes correctement et avoir une idée plus précise du fonctionnement de Kde, rendez-vous dans le répertoire /opt/kde/share/mimelnk contenant les liens Kde généraux. Vérifiez dans un premier temps que le type Mime n'existe pas déjà ; s'il en va ainsi, créez vos propres liens en prenant modèle sur les liens généraux, avec comme filtres les extensions des fichiers. Vous venez de créer un nouveau type Mime. A présent, vous pouvez l'associer à une application. Pour cela, cliquez maintenant sur "édition" et "applications". Ici, élaborez votre nouvelle application en cliquant sur "fichier", "nouveau" et "programme". De même que précédemment, il existe déjà une liste générale dans /opt/kde/share/applnk. Pour associer une application à un type Mime, sélectionnez à nouveau "propriétés", puis dans l'onglet "applications", choisissez dans la fenêtre de droite le type correspondant que vous insérerez dans la fenêtre de gauche en appuyant sur <--. Voilà, le tour est joué. Désormais, lorsque vous cliquerez sur ce type de fichiers, ils seront lus avec cette application.*

# SuSE 6.0

*La toute dernière version produite par nos voisins de Nuremberg se révèle, sous bien des aspects, séduisante et innovante. Présentée sur le stand SuSE lors du dernier Comdex à Paris, cette mouture francisée devrait être actuellement disponible.*

Le traditionnel ensemble de cinq CD, accompagné d'une disquette de boot à l'intention des systèmes ne sachant pas booter sur un CD, se voit agrémenté d'un épais manuel papier, en français ; celui-ci bénéficie en outre d'une excellente traduction. Rapidement, on boote sur le floppy ou le CD, et, aprés les messages classiques, on arrive au choix de la langue d'installation. Première surprise : entre les classiques langues allemande, française et anglaise, on trouve aussi mais pas seulement – l'espagnol, le grec, le hongrois et le russe ! ! ! Spacciba ! Lorsque l'on passe à la première phase de l'installation, à savoir le chargement des modules, les choses ne se passent pas comme on pourrait l'espérer, méme si l'on ne rencontre pas de problèmes majeur. On constate notamment quelques dérapages au moment de la détection de la carte son et des égarements lors de la détection des controlleurs SCSI. Vient ensuite la phase de sélection de la partition de swap, avec comme toujours la possibilité de ne pas reformater une partition que l'on sait valide (ce qui sauve toujours un peu de temps) et le choix du volume sur lequel on effectue l'installation ainsi que son partitionnement. Signalons pour les gens pressés ou les novices qu'il est possible de laisser au systéme le partitionnement en automatique, si toutefois on confie l'ensemble du disque à *Linux.*




Ce dernier partitionnement, quoique sommaire, se révèle efficace et permet à l'utilisateur inexpérimenté de ne pas se poser de questions. Il faut alors se pencher sur le choix des packages. *Yast* n'a pas changé d'un iota ; à ce moment précis, on doit, hors des configurations préétablies (la totale, réseau...), se plonger dans des listes sans fin de produits, et dire si oui ou non on désire les installer.

Cette méthode, qui peut paraitre rébarbative, comporte son lot d'avantages :

on contrôle trés scrupuleusement l'installation de chaque composant logiciel.

les dépendances sont gérées automatiquement, mais avec validation manuelle.

elle permet de découvrir des packages (pour ceux qui ne les connaissent pas tous, bien sûr).

Une fois les choix effectués et les conflits et dépendances réglés, il est possible d'enregistrer la configuration ainsi créée sur une disquette, afin de pouvoir dupliquer n fois un clone parfait de l'installation que l'on s'apprête à effectuer.

**Nouveautés**

Sur le plan des changements, notons que *Yast* offre désormais l'opportunité de choisir son *Window Manager.* Il faut dire que l'absence de cette option devenait de plus en plus difficile à supporter ; de plus, SuSE a très bien su intégrer cette fonctionalité désormais indispensable. L'installation achevée, sans surprises ni questions impromptues, on se logge tout de suite afin de voir ce qu'il en retourne. Si on demande le démarrage direct sous X à partir du mode console, on ne tarde pas à rester coi devant la multiplicité des environnements graphiques proposés. L'utilisateur procède à un nouveau choix et les choses sérieuses commencent... En parlant de choses sérieuses, évoquons déja celles que l'on ne voit pas de prime abord : le kernel version 2.0.36, dont la stabilité n'est plus à démontrer, répond présent, ainsi que le 2.1.130, qui ravira les possesseurs de cartes multiprocesseurs, grâce à son support SMP. La glibc intervient également ; elle remplace la libc5, mais n'a pas d'impact sur les applications utilisant cette dernière bibliothèque. La version la plus récente de *XFree86,*

*3.3.3*, apporte quant à elle le support de nouveaux matériels, comme les *Matrox G100* et *G200*, ainsi que le TNT de NVidia. Grande découverte pour les possesseurs d'ordinateurs portables, le support des chips NeoMagic est désormais intégré. *SaX* rend la configuration de l'environnement graphique toujours aussi simple, et se montre plus habile que jamais pour détecter des matériels exotiques, qu'ils soient PCI ou AGP ou qu'ils recquièrent des paramétrages de bon niveau. Le support des configurations ISDN/RNIS a désormais abouti, grâce à l'introduction du package HiSaX, et rend enfin cette solution techniquement abordable et attrayante pour tout un chacun. Ceux qui ont adopté Numéris vont sauter de joie ! La 3D est mise à l'honneur avec l'apparition des bibliothèques Glide (ils sont aussi responsables d'OpenGl pour Voodoo). Précisons enfin, toujours au sujet de l'image, que l'on retrouve ici d'innombrables logiciels, comme par exemple les ténors *Blender*, *Pov*, et *The Gimp*.

**Et pour travailler ?**

Pour ce qui touche à la bureautique, outre la version de démonstration d'*Applixware 4.4.1* (réussie, au demeurant), on dispose de... *StarOffice 5.0*, version COMPLETE et GRATUITE (dans le cadre d'une utilisation privée). Ce superbe intégré de qualité professionelle (traitement de texte, tableur, dessin vectoriel, outil de présentations, E-mail, gestionnaire de tâches et création de documents Html) apporte une réelle plus-value à la Suse. En ce qui concerne ce "monstre", le SGBD MySQL, avec ses multiples interfaces, vous permettra d'intégrer rapidement et facilement des bases de données à vos diverses applications.

Du côté des logiciels en version de démonstration, on citera, outre *ApplixWare Arkeia*, logiciel de sauvegarde bien connu, *Adabas* (celui-ci n'est ni en démo, ni en version limitée dans le temps, mais en "Personal Edition"). Les émulateurs demeurent toujours aussi nombreux. Des vieux systèmes aux consoles de jeux, presqu'aucun ne manque, excepté un ensemble d'outils destinés au support du PDA de tout linuxien qui se respecte : le Palm Pilot. Toutefois, il semblerait que les choses évoluent vite dans ce domaine.

Les jeux n'ont guère changé et on renoue avec tous les classiques du genre, sans surprise ni innovation aucune. Vivement *Civilisation 3* ! Les produits plus traditionnels, ceux dont on a une telle habitude qu'on finit par ne plus penser à eux, n'ont cependant pas subi de mise au rencart, et sont inclus dans leur dernière distribution. Ainsi, *Netscape* est livré en version *4.5 et Kde*, en *1.0* avec *Gt1.42*, *Perl 5.00502*, *gcc 2.7.2.1* et *egcs 1.1*. Dans le cadre des choses qui progressent, pas toujours aussi vite qu'on pourrait le souhaiter, mais qui avancent néanmoins, le support des processeurs Alpha devrait se voir concrétisé au début de l'été, avec une bêta au printemps. Au vu de la qualité des produits SuSE, c'est très prometteur, et donnera aux possesseurs de ces machines une alternative dans l'alternative. Sur le long terme, il devrait aussi apparaître le support des processeurs SUN SPARC et ULTRASPARC. Rendez-vous dans un prochain numéro de *Dream*, pour une étude complète de SuSE sur machine *Alpha* !

En conclusion, cette version française du seul éditeur européen de *Linux* se révèle très bien conçue et extrêmement complète. Les évolutions techniques sont innovantes tout en restant dans un domaine qui préserve la stabilité du système et de ses applications. L'internationalisation se montre excellente, tandis que les très nombreux produits et composants logiciels intégrés à cette distribution devraient combler les désirs d'une très grande majorité d'utilisateurs.

*Christophe Le Connelier*

SuSE
6.0
▶ Installation
▶ Configuration
▶ Premiers pas
SuSE LiNUX
Toutes les ressources pour votre système Linux!

*La nouvelle Suse est arrivée.*

**Suse 6.0**

*Source : http://www.suse.com*
*Statut : commercial*

En hausse

- ▲ *Support glibc*
- ▲ *Kernels 2.0 et 2.1*
- ▲ *Support ISDN/RNIS*
- ▲ *Pléthore de produits (~900 packages)*
- ▲ *Très belle internationalisation*
- ▲ *On peut payer en Euros*

En baisse

- ▼ *Quelques faiblesses en ce qui concerne la détection/chargement des modules lors de l'installation*

Note
★★★★☆

# La distribution Mandrake 5.3

*Avec cette nouvelle version, la Mandrake s'apprête à devenir un standard.*




Le PowerPack 5.3.

Axée sur la RedHat, la Mandrake a toujours passé pour sa petite sœur : une version allégée, réservée aux non-initiés du monde *Linux*. Avec le PowerPack, la Mandrake pourrait bien devenir l'aînée de la Red Hat 5.2, sur laquelle elle s'appuie. Le coffret a bénéficié d'un réel effort commercial, mais il reste hélas difficile à trouver, surtout si on n'habite pas la capitale. Le pack se compose de quatre CD, accompagnés d'un petit livret explicatif qui aidera le débutant à installer *Linux* sans problème. Ce guide ne s'adresse pas à des utilisateurs avertis, mais conviendra parfaitement aux novices. Le premier CD contient la version 5.3 de la Mandrake (festen), le deuxième fournit les sources, le troisième CD abrite utilitaires et applications au format RPM et le dernier CD accueille des applications commerciales et des démos, comme par exemple *Corel Wordperfect 8.0*. La Mandrake est également disponible sur un seul CD, pour ceux que seul le système intéresse.




**Des packages en plus**

Les packages fournis avec la Mandrake abondent. Rien n'a été retiré de la RedHat. En revanche, on trouvera *Kde*, version 1.1, ou bien encore les émulateurs wine, xemacs et licq (clône de icq), compris dans les packages de base. Les serveurs *XFree86* ont en outre bénéficié d'une mise à jour : la version 3.3.3 est ainsi livrée en standard. Quant au système, il s'agit du noyau 2.0.36. Toutefois, vous pouvez également vous rendre sur le site de la Mandrake (http://www.linux-mandrake.com) pour y télécharger les mises à jour (des fichiers rpm), qui correspondent au noyau 2.2.1 L'équipe de la Mandrake prouve de cette manière qu'elle reste très réactive par rapport aux nouveautés du monde *Linux*.

**Installation facile**

Les habitués du chapeau rouge ne seront pas dépaysés ; cette version hérite en effet des nouveautés de l'interface d'installation de la Red Hat 5.2. De plus, on a également le choix entre les trois types de mise en place : station de travail, serveur ou option personnalisée. Après avoir rendu la Mandrake opérationnelle, il ne reste plus qu'à relancer le système, à se loguer et à taper startx pour apprécier vraiment cette version. En effet, la réelle différence provient des configurations de fichiers : *Kde* est non seulement fourni, installé par défaut, mais également personnalisé. On a réglé en votre faveur de nombreux détails, comme les icônes du CD-Rom ou de la disquette, le menu de *Kde*... Bref, les adeptes de ce *Window Manager* seront sans doute agréablement surpris. Les novices de *Linux* ou de *Kde*, de leur côté, auront la joie de découvrir un environnement utilisable immédiatement et agréable d'une machine puissante et désirent faire de la bureautique ou naviguer sur le Web sans se soucier de la configuration de *Linux*. La Mandrake, par l'entremise de *Kde* et de la RedHat, a réussi son pari : élaborer une version de *Linux* facilement installable, "prête à l'emploi" et quasiment intuitive. Peut-être s'agit-il également d'un piège pour le non-initié, qui voudrait apprendre à se servir de *Linux* ; il lui faudra faire abstraction du superbe environnement graphique et des clics de souris pour s'intéresser de plus près aux fichiers de configuration. Et la Mandrake a déjà effectué tout cela à sa place.

**Et à part Kde ?**

Pour les ennemis de ce *Window Manager*, la Mandrake reste tout de même utilisable avec un autre environnement graphique (certes moins agréable), pour peu que l'on sache configurer soi-même les fichiers de démarrage de *X11*. Elle représente avant tout une version *Linux* standard, axée sur la RedHat, avec toutes ses qualités et tous ses défauts.

*Karine Mordol*




Kde déjà configuré.

**Mandrake 5.3**

*Statut : commercial*

En hausse

▲ *Xfree86 3.3.3 en standard*
▲ *Kde 1.1 configuré*
▲ *Wordperfect 8*
▲ *La documentation de la Mandrake*

En baisse

▼ *Le guide d'installation, un petit peu pauvre*

Note

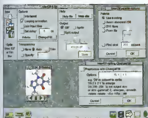
★★★★☆

# InterGif

*L'outil ultime pour créer des gifs animés.*

*InterGif* est un logiciel de création de gifs. Cela va des capacités les plus classiques aux fonctionnalités qui permettent de réduire la palette d'une image ou d'utiliser un niveau de transparence. La gestion des différentes séquences s'avère complète : on part d'une série de sprites pour obtenir une animation apte ou non à boucler et dont la vitesse est réglable. *InterGif* effectue une compression des images en fonction de celles qui les précèdent dans la scène, ce qui réduit notablement la taille du fichier final.

On dispose donc de toutes les options traditionnelles, réservées à la création d'un gif. Accessoirement, *InterGif* autorise aussi la transformation d'un gif en sprite ou d'une animation en série de sprites. Il accepte en entrée les fichiers sprites, Draw ainsi que ceux créés à l'aide du logiciel *The Complete Animator* de Iota Software. *InterGif* peut être utilisé sous *Wimp* ou via la ligne de commande, mais ne fonctionne en pratique que sur les machines possédant au moins 4 Mo de Ram, ce qui limite souvent son usage au Risc PC.


Vue d'InterGif.

Le logiciel pâtit de quelques petits bugs, comme des messages d'erreur peu explicites ; toutefois, ils ne se révèlent guère gênants à l'usage. Cela n'enlève rien aux formidables qualités de ce logiciel, qui reste un des meilleurs outils dans ce domaine. *InterGif* est disponible sur le site de l'auteur, de même que des versions pour *Windows* et *Unix* (dont une pré-compilée pour *Linux*). On y trouvera également d'autres utilitaires intéressants, tels que *AADraw*, et *ManyFonts*.

*David Feugey*

**InterGif 6.05**

*Source : www.ant.co.uk/~peter/software/intergif.htm*
*Statut : gratuit*

En hausse
▲ *Fonctionnalités disponibles*
▲ *Utilisation possible à partir de la ligne de commande*

En baisse
▼ *Quelques bugs*
▼ *Besoins en mémoire assez importants*

Note
★★★★★

# DrawLots et DrawRot

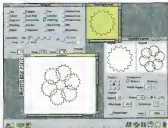
*Deux utilitaires pour mieux profiter de Draw.*

*Draw* constitue une application idéale pour créer des dessins vectoriels. Rapide et de petite taille, elle est de surcroît livrée en standard avec *Risc OS*. Malheureusement, les fonctionnalités de *Draw* demeurent en nombre restreint. *DrawLots* et *DrawRot* surgissent alors à point nommé.

Le premier de ces utilitaires permet de dessiner des figures géométriques diverses, de l'arc de cercle à la flèche en passant par l'étoile. Il est évidemment concevable de tout paramétrer, ce qui autorise la création d'un nombre infini de formes. Dans un autre registre, le but de *DrawRot* consiste à effectuer des rotations sur un fichier *Draw*. Celles-ci s'effectueront si besoin sur plusieurs copies du fichier d'origine. Ainsi, on aura loisir de disposer en cercle cinq copies d'une étoile dessinée à partir de *DrawLots*.

Ses outils s'intègrent parfaitement à *Risc OS*, le "tirer/lâcher" fonctionnant à la perfection. En outre, l'utilisateur se trouve en mesure d'exploiter aisément la sortie de *DrawLots* comme fichier source pour *DrawRot*, sans avoir à opérer une sauvegarde intermédiaire sur le disque dur.

Ces deux utilitaires gratuits sont l'œuvre de Rosemary Miskin, qui vous propose également d'autres logiciels pour gérer les fichiers Draw. Leur utilité se révèle indéniable. Ils représentent donc des compagnons appréciables pour les adeptes de logiciels de dessin vectoriel.


Echanges entre DrawLots, DrawRot et Draw.


Utilisation de DrawLots...

*David Feugey*

**DrawLots et DrawRot**

*Source : www.argonet.co.uk/users/miskin*
*Statut : freeware*

En hausse
▲ *En fait beaucoup...*
▲ *... Et bien*

En baisse
▼ *Deux petits bugs dans la gestion des fenêtres*

Note
★★★★★

# AmigaWriter

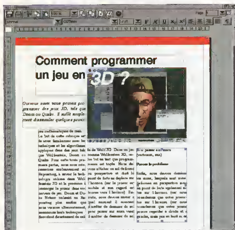
*La bureautique façon Macintosh revient sur Amiga !*

Comme son nom l'indique fortement, *AmigaWriter* est un nouveau traitement de texte pour Amiga. Développé par l'allemand Haage & Partner, ce logiciel met un point d'honneur à ne pas reprendre le fonctionnement du classique *Word pour Windows*. L'utilisateur ne dispose ni de correction en cours de frappe, ni d'édition de tableaux, ni d'outils de dessin... A n'en pas douter, les individus brouillons qui se servent d'un traitement de texte pour prendre des notes au kilomètre, en espérant les égayer avec des fonctions "pour dépanner", peuvent d'ores et déjà passer leur chemin ! Dans le petit monde totalement réinventé d'*AmigaWriter*, on renoue avec l'un des intérêts fondamentaux de l'édition de texte sur ordinateur : la présentation du document.

**Inspiré par XPress**

Contrairement à l'actuel ténor *Wordworth*, c'est bel et bien dans le monde Macintosh et non dans celui de Microsoft qu'*AmigaWriter* a été puiser son inspiration. Ici, on manipule chaque bloc de texte comme dans *XPress* : on ajuste à la souris les dimensions et l'emplacement des rectangles, en tirant sur des poignées. Si l'on souhaite obtenir une présentation en colonnes (comme dans le présent article), il suffit de créer autant de blocs de texte que de colonnes voulues et de les relier les unes aux autres (avec l'outil "chaîne", le même que celui d'*Xpress*) pour que le texte y défile de façon chronologique. La démarche se révèle aussi simple en ce qui concerne les images (grâce aux Datatypes, elles peuvent avoir n'importe quel format) ; on a moyen de les placer n'importe où dans le texte (auquel cas, le bloc de texte concerné "s'écarte" automatiquement) ou dans le fond de la page (elles remplacent alors le blanc du papier sous le texte). Fin du fin, *AmigaWriter* propose même un concept de feuilles de style "objets". On a en effet la capacité de définir des graisses, des polices, des tabulations ou encore des couleurs pour les divers ensembles du document (paragraphes, chapitres, etc.) et leur attribuer une valeur hiérarchique.

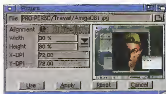
Autre point important, *AmigaWriter* est exemplaire en ce qui concerne la gestion des polices. D'une part, il reconnaît tous les formats usuels sur Amiga (bitmap et Compu graphics), sans oublier le fameux *True Type*, standard du monde *Windows*, et l'incontournable *Post Script*. D'autre part, il offre d'intéressantes possibilités, comme le réglage de l'inclinaison des italiques ou la mise en exposant et en indice des caractères. De surcroît, lorsqu'il évolue dans un document RTF, *AmigaWriter* ne se sépare pas d'une police parce qu'il ne l'a pas en stock (contrairement à *Wordworth*) : il la substitue à une autre pour la représentation à l'écran et la communique à l'imprimante pour que celle-ci fasse des recherches dans sa propre bibliothèque de fontes.




*Un traitement de texte original.*

**Quelques lacunes**

Néanmoins, il manque tout de même à *AmigaWriter* quelques options de base auxquelles tout le monde s'est habitué, tel le décompte des mots ou encore un dictionnaire français correct. Même si celui-ci répond à l'appel, il trahit tout de même un niveau relativement moyen ;




*On place une image.*

il considère les majuscules accentuées comme une faute et ne reconnaît pas les mots d'une seule lettre (tels qu'un "à" ou encore une consonne suivie d'une apostrophe). Cela dit, Haage & Partner s'est efforcé de rendre son logiciel complètement modulaire : des développeurs indépendants ne devraient pas tarder à proposer des filtres d'import/export (pour l'heure, *AmigaWriter* reconnaît l'Ascii et le RTF, mais pas les fichiers .doc de *Word*), ainsi que des modules d'outils. Verra-t-on bientôt un compteur de caractères et un éditeur de tableau ? Il n'en reste pas moins que pour l'heure *AmigaWriter* constitue la solution la plus professionnelle pour faire du traitement de texte sur Amiga, même si *Wordworth* demeure plus standard. Toutefois, méfiez-vous... En effet, bien qu'il rende la mise en page très intuitive, ce logiciel ne conviendrait pas à une PAO sérieuse, qui se satisfera nettement plus de *Pagestream*.

*Yann Serra*

**AmigaWriter 1.2**

*Statut : commercial*

En hausse

▲ *Très simple d'utilisation*

▲ *Bonne maîtrise typographique*

▲ *"Look'n'feel" à la Xpress*

En baisse

▼ *Pas de possibilité d'insérer des tableaux*

▼ *Correcteur orthographique moyen*

▼ *Absence d'outils standards (compteur, éditeur de tableaux, etc.)*

Note

★★★★☆

# WarpNote

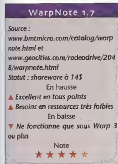
*Le roi des petites notes.*

*WarpNote* est un logiciel dont le but consiste à administrer vos notes et à supprimer ainsi l'ensemble des petits bout de papier scotchés autour de votre écran. Les outils de ce genre brillent souvent par leur médiocrité ou leur incapacité à remplacer complètement le support papier ; or, il n'en va pas de même pour *WarpNote*.

Le logiciel remplit son rôle avec brio. Chaque pense-bête est configurable à loisir, des options les plus classiques en passant par la présence éventuelle de la barre de menus ou l'affichage des notes dans le dossier des fenêtres réduites. Les positions des diverses fenêtres se voient automatiquement sauvegardées. De surcroît, la liste des notes permet d'en sélectionner facilement une. Une fonction de recherche de texte ajoute encore de l'intérêt à ce logiciel.

*WarpNote* va encore plus loin en proposant l'import /export de fichier texte par simple "tirer/lâcher" effectué à l'aide de la souris ; ainsi, on peut taper un mémorandum dans son traitement de texte pour l'intégrer plus tard dans *WarpNote*. Le logiciel ajoute même quelques raffinements esthétiques (couleur, fontes) ou pratiques (mot de passe). On a également moyen de commander le logiciel par l'entremise de Rexx, ce qui ouvre de nouveaux horizons dans le domaine du traitement des notes. Le code de cette rareté se révèle très compact et le logiciel prend un minimum de place en mémoire. En revanche, il ne fonctionnera que sous *OS/2 Warp 3* ou plus. *WarpNote* mérite largement sa place dans le dossier lancement de votre système.

*David Feugey*



**WarpNote 1.7**

*Source : www.bmtmicro.com/catalog/warpnote.html et www.geocities.com/rodeodrive/2048/warpnote.html*

*Statut : shareware à 14$*

En hausse

▲ *Excellent en tous points*

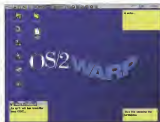
▲ *Besoins en ressources très faibles*

En baisse

▼ *Ne fonctionne que sous Warp 3 ou plus*

Note

★★★★☆



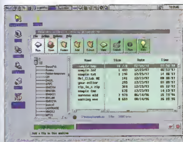

*De multiples notes.*

# WarpZip

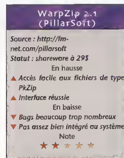
*Prometteur, mais encore imparfait.*

*WarpZip* se veut l'équivalent pour *OS/2* de *Winzip* sous *Windows*. L'installation se déroule sans problèmes ; en outre, le logiciel propose une interface et un manuel très réussis. *WarpZip* repose sur le code d'*Infozip* et gère les archives au format *PkZip*. Il demeure toutefois regrettable que les utilitaires zip.exe et unzipfsx.exe ne soient pas fournis avec l'archive. Par ailleurs, le programme dispose de fonctions internes pour visualiser les fichiers de format courant : Texte, Bmp, Gif, Pcx, Tiff, Pic et Targa. Ces utilitaires se montrent plus rapides que ceux fournis en standard avec *OS/2* et utilisés pour tous les autres types de fichiers. *WarpZip* propose également d'autres fonctions pratiques, comme l'exploitation possible d'un antivirus, la recherche dans les fichiers (Grep), le test des archives et la création d'autoextractibles. Cet utilitaire comprend cependant certaines imperfections. Ainsi, la décompression des fichiers s'opère systématiquement au même endroit du disque, les fichiers temporaires ne sont pas toujours effacés et d'autres bugs divers, notamment dans les visualiseurs intégrés, apparaissent sporadiquement. De plus, il eût fallu prévoir une intégration dans le PMShell, de même que la création d'un modèle "Archive Zip", pour offrir une simplicité de manipulation digne de ce nom. *WarpZip* constitue cependant un logiciel fort honorable ; à cet égard, le fait qu'il soit écrit spécifiquement pour *OS/2* (on n'a pas besoin ici des librairies Emx) le rend véloce. Reste à savoir maintenant si l'auteur saura corriger les nombreux défauts de sa création.

*David Feugey*



*Une interface très esthétique.*



**WarpZip 2.1 (PillarSoft)**

*Source : http://fm-net.com/pillarsoft*

*Statut : shareware à 29$*

En hausse

▲ *Accès facile aux fichiers de type PkZip*

▲ *Interface réussie*

En baisse

▼ *Bugs beaucoup trop nombreux*

▼ *Pas assez bien intégré au système*

Note

★★☆☆☆

# BeRometer

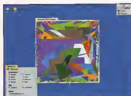
*Evaluez facilement les performances de votre machine Be.*

**B**eRometer est un logiciel de benchmark bien conçu, qui risque de faire date dans le monde *BeOS*. L'installation ne pose aucun problème particulier, mais il n'en va pas de même avec la désinstallation, qui s'avère plus qu'incomplète. Trois fois hélas, car le logiciel qui gère l'installation – *EasyInstoll* – a été créé par les mêmes auteurs. Pour le reste, la documentation se révèle simple et claire.

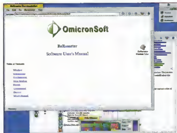
**Lancement des tests**

Les programmes utilisent souvent le multithreading, ce qui ne manque pas d'améliorer les performances de l'ensemble du système. *BeRometer* effectue une grande partie de ses tests en mode itératif et renvoie donc les performances optimales, susceptibles d'être obtenues avec *BeOS*.

Avant d'opérer les tests, il faut se placer dans le mode écran 1024 par 768, 256 couleurs et veiller à ne pas lancer d'applications pouvant freiner le système. *BeRometer* s'ouvre sur une fenêtre de sélection des tests à accomplir. Là encore, une aide en ligne répond à l'appel ; il suffit de presser Shift+F1 pour en bénéficier. L'ensemble des tests ne monopolise pas trop longtemps la machine : sur un *Pentium 200*, la durée complète de ceux-ci n'a pas excédé deux minutes.



*Test graphique.*




*Documentation de BeRometer.*

**Voici la liste des tests effectués :**

Affichage graphique : lignes, ellipses, rectangles, polygones, chaînes de caractère

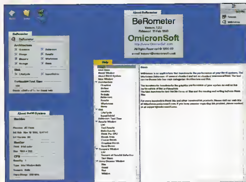
Copies et comparaisons de chaînes-Test de la mémoire

Diverses opérations mathématiques : Savage, Ackerman, Fourier, Sieve, ainsi que le calcul du Whetstone

Disque dur : création/suppression de fichiers, vitesse de lecture/écriture

Le test séquentiel et la seconde en utilisant le multitâche, ce qui mesure les performances des systèmes multiprocesseurs.

Voilà qui permet de faire le tour de sa machine. Après avoir procédé aux tests, on peut sauvegarder les résultats et les envoyer par E-mail chez les auteurs du logiciel. Il n'y a rien à craindre de ce côté, car on se trouve en face d'un simple fichier texte, qui ne comprend aucune donnée à caractère privé. Cet état de choses permet de créer une large base de comparaison pour les futurs utilisateurs du logiciel. Il faut toutefois veiller à ce que l'utilitaire *Zip* soit bien présent sur le disque pour que cette option fonctionne.




*BeRometer, vue du logiciel.*

Le logiciel propose également un tableau de comparaison bien fait. Il convient de noter les différences entre les *Pentium Mmx, Pentium II et PowerPc* G3 : sur certains résultats, le moins puissant l'emporte parfois sur son grand frère. On regrettera toutefois que le logiciel ne calcule pas un index global pour la machine. Il est effectivement assez difficile d'effectuer une comparaison entre deux machines de performances proches, puisqu'on dispose de près d'une vingtaine de résultats individuels.

**Que faire des résultats ?**

L'intérêt de *BeRometer*, en dehors de la simple curiosité qu'il se trouve en mesure de susciter, vient de la rigueur des tests. Lancé trois fois de suite, le logiciel a renvoyé des résultats identiques à chaque fois, ce qui n'a jamais rien d'évident lorsqu'on travaille sur un système multitâche. *BeRometer* a la capacité de mesurer les effets produits par un nouveau matériel sur les performances du système. On se rend compte ainsi que certaines cartes son chargent le système de façon quelque peu excessive. Dans une moindre mesure, on a aussi moyen d'évaluer l'impact d'un logiciel tournant en tâche de fond. Le graphique de comparaison donne par ailleurs de précieux renseignements sur les divers éléments du système. Le logiciel est gratuit et les auteurs restent ouverts à toute proposition concernant de nouveaux benchmarks, dont leur rejeton pourrait profiter.

*David Feugey*

**BeRometer**

*Source : www.OmicronSoft.com*
*Statut : freeware*

En hausse

▲ *Permet de tester les capacités de son système, fiable et rapide*

En baisse

▼ *La désinstallation approximative, pas d'index général de comparaison*

Note

★★★★☆







Nos Packard Bell sont livrés avec des écrans de 38 ou 43 cm en ***Multimédia.***

| Référence | Prix usuel | Prix ADFI |
|---|---|---|
| MM 333 ac | ~~6090 F~~ | 5 790 F |
| MM 366 ac | ~~6690 F~~ | 6 390 F |
| MM 400 ac | ~~7690 F~~ | 7 190 F |
| MM 350 CD | ~~7590 F~~ | 7 190 F |
| MM 350 | ~~8090 F~~ | 7 590 F |
| MM 400 | ~~9690 F~~ | 9 190 F |
| PL 350 v2 | ~~9690 F~~ | 9 190 F |
| PL 400 v2 | ~~12690 F~~ | 11 990 F |
| PL G200 | ~~14690 F~~ | 13 995 F |

| SIMM EDO 32 bits 60 ns | |
|---|---|
| 8 MO | 200 F |
| 16 MO | 400 F |
| 32 MO | 740 F |
| 64 MO | 1380 F |
| 128 MO | 3200 F |

*A.D.F.I. ARRIVE SUR PACKARD BELL*

*Une solution matérielle complète pour mieux maîtriser Linux.*

# A.D.F.I. Application

47 avenue de la libération
F63000 Clermont Ferrand.

☎ 04 73 34 34 34.

**Pour commander :**
- Passez au magasin ou,
- Écrivez-nous en joignant le paiement par chèque ou mandat cash ou,
- Téléphonez-nous avec votre carte bancaire

**Les livraisons à domicile :**

| | |
|---|---|
| Port total en France sans matériel | 50 |
| Port total en France s'il y a au moins un matériel | 80 |
| Traitement prioritaire | +20 |
| Livraison sur une île métropolitaine | +70 |
| DOM-TOM | Prix variable selon le poids et la destination |
| Zone Euro et Suisse | Port postal +15 |
| Recommandation | +70 |
| Assurance de transport par tranche de 2000 F en France | +30 |

## ACHAT - VENTE - RÉPARATION - CONSEIL

| | |
|---|---|
| Paris mon Mac | 1690 |
| Art eff. VF | 1340 |
| Art Effect | 610 |
| Computer Graphics VF (Sortie vers juin) | 690 |
| DrawStudio VF | 790 |
| Devpac 3.18 VO | 290 |
| Devpac 3.50 VF | 695 |
| Décollages VF | 290 |
| > Thème Espace | 140 |
| > Thème Lettres | 140 |
| > Thème Chiffres | 140 |
| > Thème La mer | 140 |
| Décollages complet (économisez 140 F) | 710 |
| Magic Synthesis VF (Sortie vers juin) | 690 |
| PageStream Amiga VF nouvelle version | 1690 |
| > Extension TextFX (effets de texte) | 490 |
| > Extension Borders (effets d'ornementation) | 490 |
| > Filtre JPEG | 240 |
| > Filtre WordWorth nouvelle version | 240 |
| > Effet de Grey (effets d'image) | 290 |
| > Moteur True Type (fontes .ttf) | 290 |
| > Scénario ARexx | 240 |
| PageStream complet (économisez 720 F) | 2940 |
| TypeSmith VF | 490 |
| Répondeur A.B.E. VF | 470 |
| Spouleur d'impression VF | 120 |
| Pilotes Stylus-Canon VF | 190 |
| Tome 1 Exec VF | 320 |
| Tome 2 matériel VF | 320 |
| Bien débuter A600 VF | 190 |
| Bien débuter A1200/A4000 VF | 190 |
| Clavier accentué VF | 120 |
| Environnement Workbench VF | 120 |
| 3D Images | 155 |
| Amiga Forever | 340 |
| Amiga Format 36 (et moins) | 49 |
| Amiga tools 8 (et moins) | 129 |
| Aminet 29 (et moins) | 89 |
| Aminet set 7 (et moins) | 199 |
| AmiTCP 6 (et moins) | 89 |
| Asim CDFS | 480 |
| CD Developer | 130 |
| Cybergraphx v4 | 140 |
| Cygnus ED | 220 |
| DigiBooster Pro | 290 |
| Home Dream | 440 |
| Euro CD 3 (et moins) | 129 |
| Envoy 3 | 330 |
| Fusion Mac Emulator | 699 |
| Geek Gadget | 99 |
| Gateway chaque module | 200 |
| Get connected | 440 |
| GVP | 495 |
| Hard Drive Menu | 119 |
| Ibrowse VF | 398 |
| In to the net | 135 |
| Kara collection | 265 |
| Light ROM 6 (4 CD) | 280 |
| Magic WB | 109 |
| Magic Publisher | 189 |
| Mini CD DAO | 380 |
| Meeting Pearls | 69 |
| Miami | 279 |
| Mod's Anthology | 159 |
| Mogel | 139 |
| Network offline | 39 |
| Net Connect | 580 |
| Opus Plus | 185 |
| Organiser VF | 240 |
| Personal Paint | 189 |
| PFS2 | 330 |
| Post Studio | 289 |
| Red Hat Linux | 129 |
| Recettes la compilation ARexx | 230 |
| Scala MM400 | 499 |
| Scanquix VF - Scanémulation | 690 |
| Samson System | 239 |
| Speccy Classic 99 | 190 |
| STFax Pro | 690 |
| Studio Professionnal 2.2 | 256 |
| SuperView | 199 |
| The Best of Mecomp | 119 |
| TCP | 240 |
| Tornado 3D | 2850 |
| True 3 D | 150 |
| Turbo Calc VF | 490 |
| TurboPrint | 449 |
| Ultimate Bitz Basic CD | 220 |
| Web II | 155 |
| Wildfire | 890 |
| WordWorth VF | 490 |
| Workbench Designer 2 | 115 |
| Screen Savers volume 2 | 95 |

| | |
|---|---|
| After Burner Aerial | 159 |
| Alcatraz | 159 |
| Alien 3 | 140 |
| Amiga Olofight | 269 |
| Amore Babes Le nu au féminin | 240 |
| Ballistix | 159 |
| Bills Tomato Game | 159 |
| Blade | 269 |
| Bob's Bad Day | 140 |
| Captain Blood | 140 |
| Cartoon Clipart | 175 |
| Chase HQ | 140 |
| Civilization | 220 |
| Clown-O-Mania | 140 |
| Compilation Amiga Classix (24 jeux) | 215 |
| Compilation Dream Team (3 jeux Ocean) | 140 |
| Compilation Fox Collection (4 jeux Titus) | 140 |
| Compilation Game Attack (23 jeux) | 149 |
| Compilation Metalic Power (3 jeux Lankhor) | 140 |
| Compilation Monde Merveille (3 jeux Ocean) | 140 |
| Compilation Adventure (4 jeux Megastar) | 99 |
| Compilation Fighters (4 jeux Megastar) | 99 |
| Compilation Sport 2 (3 jeux Loriciel) | 140 |
| Compilation Vulcanology (9 jeux Vulcan) | 229 |
| Crazy Cars 3 | 140 |
| Cytron psygnosis | 169 |
| Darkman | 140 |
| Eat the Whistle (football) | 215 |
| Elf | 140 |
| Euro League Manager | 180 |
| F-19 Stealth Fighter | 180 |
| FIFA International Soccer | 220 |
| Final Odyssey | 330 |
| Fire Force | 140 |
| Flying High | 189 |
| Flying High data disk | 140 |
| Forest Dump | 140 |
| Formula 1 Ferrari | 170 |
| Formula 1 Grand Prix | 220 |
| Foundation CD | 289 |
| Gateway 3 | 75 |
| G.I.G.N Opération Jupiter | 140 |
| Genetic Species | 269 |
| Gigancid | 140 |
| Golden demos | 95 |
| Guerrde P99 | 159 |
| Harlequin | 150 |
| Hook, le capitaine Crochet | 140 |
| Imperator | 150 |
| Infestation | 140 |
| Jet | 140 |
| Knights of the Sky | 170 |
| Kung Fu | 380 |
| Krusty Fun House les Simpsons | 140 |
| Lemming | 140 |
| Lemmings II | 169 |
| Les aventures de Moktar | 140 |
| Lethal Weapon | 140 |
| Lotus III | 140 |
| McDonaldland | 140 |
| Mickey 123 | 140 |
| Mickey ABC | 150 |
| Mickey Mémoire | 180 |
| Mickey Puzzles animés sur A500 | 140 |
| Mickey 123 + abc + Mémoire + Puzzle | 560 |
| Mogel | 140 |
| Motorhead | 140 |
| Myst CD ou disquette | 379 |
| Navy seals | 140 |
| Nemac IV | 140 |
| Nigel Mansell | 140 |
| No Second Prize | 140 |
| Olofight | 229 |
| On escapee | 259 |
| Othello Killer | 140 |
| Piracy on the High seas | 140 |
| Push-Over | 140 |
| Quake CD | 339 |
| Robocop | 140 |
| RVF Honda | 140 |
| Samba World Cup | 269 |
| Scene Xplorer2 | 145 |
| Silent Service II | 140 |
| Silk Worm | 140 |
| Sim City | 140 |
| Simant - les fourmis | 140 |
| Six Sens Investigation | 359 |
| Sky Blaster | 140 |
| Star Wars | 140 |
| Street Fighter 2 | 140 |
| Super Hang On | 140 |
| Super Tetris | 140 |
| Suspicious Cargo | 150 |
| Sword of Sodan | 140 |
| Theme park | 140 |
| The scene archive | 145 |
| Thunder Blade | 140 |
| Time Keepers | 189 |
| Time of Reckoning | 235 |
| Tiny Sweek | 140 |
| UFO | 140 |
| Ultimate Skidmarks | 159 |
| Urepa | 189 |
| Virtual Karting | 180 |
| Vulcan Bonanza | 239 |
| Vroom data disk | 140 |
| Wild Wheels Karting | 140 |
| Wing Commander | 265 |
| Wizkid | 140 |
| Zombie Massacre | 165 |

**Matériels spécifiques Amiga**

| | |
|---|---|
| Ordinateur Amiga A1200 Magic | 2590 |
| Ordinateur Amiga A1200 Magic HD | 3290 |
| Ordinateur Amiga A1300 Tour | 3340 |
| Ordinateur Amiga A1400 Tour Zorro II | 4849 |
| Ordinateur Amiga A1500 Tour Zorro III | 5949 |
| Apollo 630 68030 33 Mhz + Copro | 800 |
| Apollo 630 68030 50 Mhz + Copro | 1150 |
| Apollo 1240 68040 25 Mhz + Copro | 1390 |
| Apollo 1240 68040 40 Mhz + Copro | 1990 |
| Apollo 1260 68060 50 Mhz + Copro | 2990 |
| Apollo 1260 68060 66 Mhz + Copro | 3590 |
| Apollo 2030 68030 50 Mhz + Copro + SCSI | 1790 |
| Apollo 4040 68060 40 Mhz + Copro + SCSI | 2490 |
| Apollo 4060 68060 50 Mhz + Copro + SCSI | 3850 |
| Apollo 4040 68040 66 Mhz + Copro + SCSI | 4490 |
| Ariadne Ethernet version complète non allégée | 890 |
| Blizzard 1260 | 3690 |
| Blizzard SCSI Kit | 690 |
| Blizzard PPC 603e+ SCSI 180 Mhz 68040 | 2690 |
| Blizzard PPC 603e+ SCSI 200 Mhz 68040 | 3180 |
| Blizzard PPC 603e+ SCSI 240 Mhz 68040 | 3680 |
| Blizzard PPC 603e+ SCSI 160 Mhz 68060 | 5280 |
| Blizzard PPC 603e+ SCSI 200 Mhz 68060 | 5380 |
| Blizzard PPC 603e+ SCSI 240 Mhz 68060 | 5680 |
| BVision 8MO pour Blizzard PPC | 1480 |
| Carte mémoire+horloge avec support 8 Mo | 390 |
| Cyberstorm MK3 UWSCSI 68060 | 4480 |
| Cyberstorm PPC UWSCSI 160 Mhz | 4380 |
| Cyberstorm PPC UWSCSI 200 Mhz | 5290 |
| Cyberstorm PPC UWSCSI 233 Mhz | 5780 |
| Cyberstorm PPC UWSCSI 180 Mhz | 4905 |
| Cyberstorm PPC UWSCSI 200 Mhz 68040 | 5805 |
| Cyberstorm PPC UWSCSI 233 Mhz 68040 | 5840 |
| Cyberstorm PPC UWSCSI 180 Mhz 68060 | 6880 |
| Cyberstorm PPC UWSCSI 200 Mhz 68060 | 7780 |
| Cyberstorm PPC UWSCSI 233 Mhz 68060 | 6880 |
| Cybervision 8 MO pour Cyberstorm | 1580 |
| Digitaliseur audiophonique 8 bits | 350 |
| Interface Midi professionnelle | 310 |
| Doubleur de fréquence | 659 |
| Lecteur de disquette interne DD standard | 270 |
| Lecteur de disquette interne HD standard | 589 |
| Lecteur de disquette externe HD Micronik | 629 |
| OS 3.1 VF complet A500/600/2000 | 390 |
| OS 3.1 VF complet A1200/3000/4000 | 880 |
| Parnet Amiga vers Amiga | 165 |
| Picasso IV | 2490 |
| > Concerto Picasso | 390 |
| > Paloma Picasso | 790 |
| > Pablo Picasso | 990 |
| Pixel64 | 1890 |
| Sernet Amiga vers CD32 avec lecteur | 249 |
| Souris 2 boutons | 159 |
| Souris 3 boutons haute résolution | 229 |
| Joystick | 159 |
| Topolino pour A2000 | 320 |
| Topolino autre Amiga | 295 |
| Tour Atéo | 990 |
| Tour S.D.L.I - prête à l'emploi | 1390 |
| Tour Infinitiv | 1240 |
| > Alimentation interne Micronik | 380 |
| > Coque clavier Amiga+adaptateur+cordon | 380 |
| > Carte Zorro II Micronik | 1390 |
| > Carte Zorro III Micronik | 3189 |
| > Sortie externe SCSI Micronik | 160 |
| > Adaptateur PCMCIA Micronik | 245 |
| > Lecteur interne DD Micronik | 380 |
| > Lecteur interne HD Micronik | 559 |

**Clavier PC**

| | |
|---|---|
| Clavier PS2 | 89 |
| Clavier standard | 189 |
| Clavier AT | 79 |
| Clavier multimédia Méga Tush | 149 |
| Clavier Microsoft Élite | 259 |
| Clavier ergonomique pour tour Amiga | 389 |

**Disque dur**

| | |
|---|---|
| 540 MO IBM 2½ garanti 6 mois | 1040 |
| 1600 MO IBM 2½ garanti 36 mois | 1790 |
| 3200 MO IBM 2½ garanti 36 mois | 2290 |
| 3200 MO UDMA 33 | 909 |
| 4300 MO UDMA 33 | 1099 |
| 6400 MO UDMA 33 | 1249 |
| 6400 MO IBM UDMA garanti 40 mois | 1729 |
| 8400 MO UDMA 33 | 1639 |
| 4300 MO SCSI | 2359 |
| 6400 MO SCSI UW | 2229 |
| 6400 MO IBM SCSI garanti 36 mois | 2395 |
| 9400 MO SCSI UW | 4239 |

**Graveurs**

| | |
|---|---|
| Chariot 5¼ 3½ | 29 |
| Graveur Mitsumi 2x 8 x IDE | 1649 |
| Graveur Panasonic 4x 8x SCSI | 2219 |
| Graveur Yamaha 4-4 16 SCSI | 2690 |
| Cédédrom vierge Kodak ou Verbatim | 15 |

**Imprimantes**

| | |
|---|---|
| Canon LBP 660 | 2509 |
| HP 722 C | 2329 |
| HP 6L Laser | 3219 |
| HP 690 C | 1949 |
| HP 895 Cxi | 2829 |
| Stylus color 440 | 990 |
| Stylus color 640 | 1490 |
| Stylus color 740 | 1790 |
| Toner HP Laser série 4 | 719 |
| Toner HP série 5 | 549 |
| Cable imprimante | 49 |

**Lecteurs**

| | |
|---|---|
| Lecteur cédérom 40x IDE Atapi | 590 |
| Lecteur cédérom 32x SCSI | 789 |
| Lecteur Iomega | 149 |
| Lecteur Jazz 2 GO | 3119 |
| Lecteur LS 120 | 759 |
| Lecteur ZIP | 729 |
| Disquette DD 880 KO SONY par 10 | 30 |
| Disquette 120 MO | 109 |
| Disquette Jazz 1 GO | 719 |
| Disquette Jazz 2 GO | 719 |
| Disquette ZIP | 89 |

**Modems**

| | |
|---|---|
| Modem externe Numéris | 3519 |
| Modem externe Olitec | 1179 |
| Modem externe Olitec Smart memory | 1509 |
| Modem externe USR K056 | 1449 |
| Modem externe USR K128 | 3495 |
| Modem externe USR Pro | 1679 |

**Moniteurs**

| | |
|---|---|
| 14 pouces (35 cm) | 949 |
| 15 pouces (38 cm) | 1209 |
| 15 pouces (38 cm) garanti 3 ans sur Amiga | 1209 |
| 17 pouces (43 cm) | 2119 |
| 17 pouces (43 cm) garanti 3 ans sur Amiga | 2790 |
| 19 pouces (48 cm) | 4709 |
| Adaptateur SVGA - Amiga | 180 |
| Prise Péritel - Amiga | 240 |
| Distributeur 4 écrans | 1129 |
| Basculeur 2 écrans | 179 |

| | |
|---|---|
| Epson GT 7000 | 1790 |
| Epson GT 7000 Photo | 2290 |

**Souris PC**

| | |
|---|---|
| Souris 3 boutons Luxe | 89 |
| Tapis souris | 9 |

**SCSI**

| | |
|---|---|
| Boîtier SCSI 2 emplacements | 369 |
| Adaptec 2940 Ultra wide | 1899 |
| Contrôleur SCSI | 489 |

**Connectique**

Votre adaptateur Amiga ou PC est disponible
Votre câble Amiga ou PC est disponible

**Réseau**

| | |
|---|---|
| Carte réseau Bus 100 Bnc | 109 |
| Carte réseau PCI | 119 |
| HUB 8 ports | 709 |
| Pince à sertir | 379 |
| Pince RJ45 | 319 |
| Répartiteur 8 moteurs | 1729 |
| Routeur Syxel Prestige 100 | 3769 |
| Routeur Syxel Prestige 128 | 4469 |

**Divers**

| | |
|---|---|
| Palm Pilote | 2389 |
| Philips Nino | 3259 |
| Rack IDE | 109 |

## Le coin du bricoleur PC

**Alimentation**

| | |
|---|---|
| Alimentation AT | 219 |
| Alimentation ATX | 369 |
| Cable d'alimentation Europe-secteur | 19 |
| Cable d'alimentation M/F Europe | 19 |
| Doubleur d'alimentation 5¼ | 19 |
| Onduleur 1000 VA | 1299 |
| Petite alimentation de portable | 9 |
| Réducteur 5¼ 3½ | 9 |

**Boîtier**

| | |
|---|---|
| Desktop AT | 339 |
| AT moyen | 219 |
| AT mini | 219 |
| ATX grande tour | 789 |
| AT Grande tour | 789 |
| ATX moyenne tour | 369 |

**Carte mère PC**

| | |
|---|---|
| ABIT PII 440BX | 929 |
| CM 100 Mhz S7 AGP | 459 |
| CM PII Bx ALI + CS | 509 |
| CM Bx Asustec PII | 1379 |
| CM LX PII + CS | 729 |
| ASUS 440BS 5 SCSI2 | 2509 |
| P2L97 Asustec LX Atx + AGP PII | 1109 |

**Carte graphique PC**

| | |
|---|---|
| Extension ATI 4MO | 249 |
| ATI Xpert@Works 4 MO AGP | 579 |
| ATI Xpert@98 8 MO AGP | 509 |
| ATI Xpert@works 4 MO PCI | 609 |
| S3 1 MO ext 2 MO | 169 |
| Carte MIRO DC 30+ | 8975 |
| Carte TV MIRO PC TV | 819 |
| SIS 3 D AGP 4 MO + TV | 269 |
| SIS 8 MO AGP | 229 |
| Vision 1 | 238 |
| Intel i740 8 MO AGP | 509 |
| S3 4 MO | 259 |
| TRIDENT AGP | 259 |
| N°9 AGP 4 MO SWRAM | 289 |
| Millenium G200 AGP 8 MO | 789 |
| VOODOO II 3DX 12 MO | 1339 |

**Carte son PC**

| | |
|---|---|
| Carte compatible Sound Blaster | 109 |
| Carte Sound Blaster 16 | 239 |
| Carte Sound Blaster 64 PCI | 229 |
| Carte Sound Blaster 128 PCI | 619 |
| Haut parleur 60 W | 69 |
| Haut parleur 240 W | 179 |
| Haut parleur 300 W | 479 |
| Haut parleur 500 W | 609 |
| Haut parleur Proview | 148 |
| Live 128 | 1909 |
| Micro casque | 49 |

**Lecteur DVD**

| | |
|---|---|
| Lecteur DVD ROM | 1099 |
| Lecteur DVD ROM | 2249 |

**Logiciels PC**

| | |
|---|---|
| Ciel compta | 1479 |
| Commando | 439 |
| Coupe du monde 98 | 239 |
| Encyclopédie Hachette 99 | 679 |
| Family pack | 239 |
| Grand prix | 339 |
| Jeux Sega | 59 |
| Microsoft Family Pack | 239 |
| Microsoft office étudiant | 949 |
| Office PME | 2393 |
| Pack Découverte du multimédia | 249 |
| Pack Office Lotus Smart Suite | 79 |
| PC Anywhere | 1419 |
| Via Voice 98 exclusive | 1479 |
| Win 98 OEM | 819 |
| Works Suite 99 | 1059 |

**Mémoire SDRAM**

| | |
|---|---|
| SD RAM 032 Mo 100 Mhz | 369 |
| SD RAM 064 MO 100 Mhz | 719 |

**Modem PC**

| | |
|---|---|
| Modem PCMCIA USB 33,6K | 1209 |
| Modem interne Eurovivs | 209 |
| Modem interne ISDN | 619 |
| Modem interne Kortex | 399 |
| Modem interne Olitec 056K | 459 |
| Modem interne USR 056K | 529 |
| Video Splitter | 179 |

**Processeurs PC**

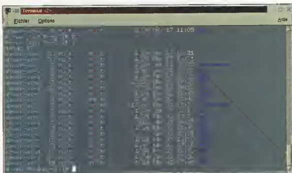
| | |
|---|---|
| Celeron 333 A | 639 |
| Celeron 400 A | 1489 |
| Cyrix 200 | 639 |
| K6 2 333 | 1019 |
| K6 2 350 | 1129 |
| MII 300 | 549 |
| Pentium II 333 | 1889 |
| Pentium II 350 | 1729 |
| Pentium II 400 | 2809 |
| Ventilateur Celeron | 69 |
| Ventilateur PII | 49 |
| Ventilateur pour boîtier | 69 |
| Ventilateur S7 | 29 |

**Scanners**

| | |
|---|---|
| Scanner parallèle 300 x 600 | 409 |
| Scanner SCSI 300 x 600 | 1019 |
| Tableau graphique Trust | 1659 |

# Les fichiers sous Linux

*/Root, /etc, /bin... constituent autant de répertoires parfois obscurs. Pourtant, les fichiers sous Linux ne sont pas classés n'importe comment et les répertoires se révèlent plus parlants qu'on ne pourrait le croire.*



*Les fichiers à la racine.*

**L***inux* a repris la philosophie *Unix*, qui considère chaque chose comme un fichier, qu'il s'agisse des périphériques tels que les lecteurs de disquettes, des commandes comme mkdir ou ls, des fichiers de configuration ou bien encore des répertoires. Tout cela est représenté sous la forme d'un vaste système de fichiers.

**Mais comment se repérer ensuite dans ce labyrinthe ?**

Le tout premier fichier à connaître s'appelle "/". Il s'agit de la racine. On ne peut remonter au-delà, car elle représente la base de l'arborescence du système. Il existe bien une commande pour chercher où se situe un fichier (find), mais il est quand même plus pratique d'avoir déjà une idée de l'emplacement du fichier cherché. Les noms des répertoires peuvent vous y aider. Ils n'ont pas été choisis au hasard et il suffit bien souvent de se souvenir simplement de leur signification.

**Les commandes dans bin**

Bin comme binaire... ces répertoires contiennent les exécutables en binaire des commandes utilisables. C'est d'ailleurs pour cela que chaque utilisateur doit posséder dans son PATH les chemins complets de ces répertoires. Tapez echo $PATH pour voir ce que contient le vôtre. Le répertoire /bin renferme les commandes essentielles, comme cp ou ls par exemple. Le pendant de ce répertoire se nomme /usr/bin et comprend en quelque sorte les commandes qui ne sont pas indispensables au démarrage. Le répertoire /sbin (de même que le répertoire /usr/sbin) recèle des commandes plus spécifiques à l'administrateur, telles que mke2fs, afin de formater une partition au format *Linux* et *Fdisk*, réservée au partitionnement du disque. Les exécutables graphiques se situent quant à eux dans le répertoire /usr/X11R6/bin.

**Les fichiers C dans lib et include**

Les lib abritent des librairies, appelées aussi bibliothèques du langage C utilisées par les programmes (ou par les commandes, ce qui pour *Linux*, revient au même). Le répertoire /lib accueille les bibliothèques communes (shared librairies). Chaque programme peut ensuite posséder ses propres bibliothèques, ce qui explique la présence de nombreux sous-répertoires lib. On trouvera également plusieurs fois des répertoires appelés include, qui comportent des fichiers d'en-tête en c (les .h). Le répertoire qui contient les en-têtes standards communs à tous se nomme dans /usr/include.

**Et l'aide ?**

Les répertoires man comprennent l'aide en ligne, que l'on obtient en tapant la commande man. Il existe également des répertoires doc, qui comme leur nom l'indique abritent de la documentation. Les Howto résident dans le répertoire /usr/doc.

**Boot : fichiers du démarrage**

En général, les distributions de *Linux* (entre autres, RedHat) incluent ce répertoire. A l'intérieur cohabitent le noyau, vmlinuz suivi du numéro de noyau (vmlinuz-2.0.36-0.7 pour le noyau standard de la RedHat 5.2) et les fichiers nécessaires au démarrage de *Linux*. C'est dans ce répertoire que va lire *Lilo* pour amorcer le système.

**Dev et les périphériques**

*Linux* administre les périphériques comme des fichiers. Ils séjournent dans ce répertoire. Ainsi, /dev/psaux gère la souris, /dev/cdrom contrôle le CD-Rom et /dev/fd0 s'occupe du lecteur de disquettes. Les fichiers hda, hdc... gouvernent les disques et hda1, hda2... les partitions. Les écrans ou consoles virtuels sont régis par les fichiers tty. En pratique, on visite rarement ce répertoire. Si vous voulez accéder au CD-Rom, par exemple, il faudra se placer dans le répertoire où celui-ci est monté, et non pas dans /dev/cdrom ou /dev/hdb.

**Les fichiers de configuration : etc. !**

Dans /etc se trouvent les fichiers de configuration générale, tels que passwd, qui contient la liste des utilisateurs et leur mot de passe, fstab, qui indique à la commande de mount comment une partition ou un lecteur doit être monté, ainsi que smb.conf, le fichier de configuration de samba. Ici, il s'agit de fichiers généraux et non pas des configurations propres à chaque utilisateur, celles-ci se situant dans leur home respectif sous la forme .config. Dans /etc, des répertoires regroupant des commandes et des fichiers de configuration classés par thème la gestion du réseau par exemple, ou bien encore le répertoire ppp pour configurer les liaisons par modem – se tiennent également à la disposition de l'utilisateur. Le répertoire rc.d a une grande importance : il contient les scripts de lancement du système. Il se divise lui-même en sous-répertoires, comme par exemple init.d, lequel se compose des scripts des modules lancés au démarrage. Le répertoire /etc/X11 renferme

Un exemple d'arborescence Linux.

pour sa part les fichiers de configuration de l'environnement graphique. Le *XF86Config*, qui recèle les informations de configuration du matériel, y a élu domicile. /etc/skel est également un sous-répertoire majeur : il comprend les fichiers que chaque nouvel utilisateur se verra attribuer par défaut dans son home, si celui-ci est créé grâce à l'utilitaire linuxconf.

**Home et root : chacun chez soi**

Si votre système bénéficie d'une bonne gestion, /home comporte autant de sous-répertoires que d'utilisateurs. Chaque répertoire portera le nom de l'un d'entre eux, ce qui explique que l'on emploie couramment l'expression "se placer dans son home" Lorsqu'un utilisateur se connecte, il est logé tout de suite "dans son home". L'utilisateur toto par exemple réside dans /home/toto : il est chez lui. L'utilisateur root possède quant à lui un home particulier, /root, qui se situe en-dehors de /home. Chacun est alors libre de configurer son home comme il l'entend et d'y gérer les droits d'accès. L'utilisateur reste véritable propriétaire de cette partie de l'arborescence. Seul le root pourra outrepasser les permissions d'accès.

**/mnt : mais où est le CD-Rom ?**

Plutôt que de créer des répertoires à la racine pour le CD-Rom ou encore une partition DOS, l'administrateur préférera élaborer ces répertoires dans un répertoire appelé mnt pour mount. Si on veut parcourir le contenu d'un CD-Rom, après l'avoir monté bien sûr, il faudra se placer dans le répertoire /mnt/cdrom. Pour une partition DOS, il s'agira éventuellement de /mnt/dos... Afin de parcourir la liste des points de montage qui se situent dans ce répertoire, il suffit de taper "ls" dans le répertoire mnt. Enfin, tapez la commande mount, sans argument, si vous souhaitez savoir quels sont les partitions ou lecteurs montés.

**Les processus de proc**

/proc constitue un répertoire un petit peu particulier. A l'intérieur séjournent tous les renseignements concernant les processus actifs. On y trouve ainsi des répertoires, dont le nom correspond au numéro de processus auxquels ils font référence. Ce n'est pas réellement un système de fichiers classique, mais il s'avère très utile pour connaître l'état d'un processus ou bien encore les ressources de la machine.

**/usr : le supplément**

/usr accueille les ressources supplémentaires du système. Ainsi, les répertoires /usr/bin, /usr/sbin et /usr/lib ont la même signification que /bin, /sbin, /lib, mais contiennent des commandes et bibliothèques qui ne sont pas indispensables au fonctionnement du système en mode réduit. /usr/lib/X11 renferme quant à lui les librairies propres à l'environnement *X11* et le répertoire /usr/X11R6 regroupe tout ce qui concerne directement *X11*. Ceux qui voudront recompiler leur noyau s'intéresseront également au répertoire /usr/src, qui abrite les sources du système. A l'intérieur de /usr se trouve également le répertoire local, dont le découpage suit le même principe que /usr. Il aura deux usages distincts : dans un réseau, il permet de différencier le répertoire /usr général, installé sur le serveur, du répertoire /usr/local, installé sur votre machine. Pour les ordinateurs qui ne sont pas en réseau, ce répertoire sert très souvent à l'installation de nouveaux programmes. On peut ainsi décompresser les sources dans src, mettre les exécutables dans bin, la documentation dans man... Ce principe permet alors de ne pas avoir à réinstaller ses programmes, lorsqu'on change de version. Il donne également le privilège de faire fonctionner plusieurs systèmes *Linux* avec les mêmes logiciels, installés une seule fois. Pour cela, il suffit d'attribuer une partition à /usr/local, qui sera utilisable par les différentes versions *Linux*.

**Var, les fichiers qui bougent**

Dans /var (pour variable) résident par exemple le spool de l'imprimante, un cache et tous les fichiers qui sont par nature amenés à être modifiés en permanence.

*Karine Mordal*

**Tableau récapitulatif**

| *Répertoires* | *Contenu* |
|---|---|
| */* | *La racine* |
| */boot* | *Le noyau et les fichiers de démarrage* |
| */bin* | *Les commandes (en binaire)* |
| */sbin* | *Des commandes supplémentaires (en binaire)* |
| */lib* | *Les librairies communes* |
| */dev* | *Les périphériques* |
| */mnt* | *Les partitions et les lecteurs* |
| */home* | *Les répertoires des utilisateurs* |
| */root* | *Le répertoire "home" du root* |
| */proc* | *Les processus* |
| */var* | *Les fichiers dont le contenu change souvent* |
| */tmp* | *Les fichiers temporaires* |
| */etc* | *Les fichiers de configuration générale avec des sous-répertoires à thèmes* |
| */etc/X11* | *Les fichiers de configuration de X11* |
| */usr/X11R6* | *Les principaux fichiers et commandes concernant X11* |
| */usr/X11R6/bin* | *Commandes binaires graphiques* |
| */usr/bin* | *Commandes des applications* |
| */usr/doc* | *Documentation sur Linux* |
| */usr/include* | *En-têtes communs* |
| */usr/src* | *Les sources du noyau* |

# Se retrouver dans BeOS

*Voici la présentation de l'arborescence de BeOS.*

Prenez un *Unix*, enlevez la gestion multiutilisateur et vous obtenez à peu de choses près l'arborescence de *BeOS*. D'un côté, nous trouvons *Unix*, qui range tous ses fichiers dans des répertoires nombreux, mais (relativement) explicites. En face, *DOS/Windows* et quelques autres utilisent la solution exactement opposée et placent tous les fichiers du système dans une poignée de répertoires surchargés. *BeOS* constitue une sorte de compromis, puisqu'il offre une arborescence *Unix* largement amputée.

**Les répertoires de BeOS**

- La racine / :

Il s'agit bien évidemment du répertoire parent de tous les autres répertoires.

- Le système "réel" : /boot

Il représente l'emplacement "physique" de tous les répertoires systèmes de *BeOS*. Toutefois, pour plus de commodité, des liens depuis la racine sont créés.

- Les commandes shell : /bin (pointe sur /boot/beos/bin)

Ce répertoire contient l'ensemble des commandes système que vous pouvez invoquer depuis le shell. Il ne s'agit donc pas des applications graphiques. Toutefois, des outils précieux séjournent ici, comme le compilateur c++ (gcc) ou le compresseur gzip. Signalons un point essentiel : la commande beep, qui fera rugir votre carte son, répond à l'appel !

Les fichiers de périphériques : /dev

Sous *Unix*, les périphériques sont perçus comme des fichiers contenus dans le répertoire /dev. Sous *BeOS*, la présence de ce répertoire est principalement expliquée par la compatibilité recherchée avec les systèmes *Unix*.

Le système : /system (/boot/beos/system)

Dans ce répertoire, on trouve le code (objet) du noyau de *BeOS*, le programme *DeskBar* (barre des tâches) et le Tracker.

> **Les liens symboliques**
>
> *Les liens symboliques, très utilisés sous Unix et donc sous BeOS, sont des fichiers "vides" qui pointent vers d'autres fichiers. Il s'agit d'autres liens, des fichiers réels ou des répertoires. Ces indirections rendent la compréhension du système de fichier un peu plus complexe, puisqu'un unique fichier réel ou un répertoire peut être référencé à différents endroits. On utilise souvent cette technique, afin de ménager des raccourcis : BeOS s'en sert parfois pour créer des icônes, qui vous permettent d'aller droit au but sans encombrer votre écran de fenêtres "intermédiaires".*

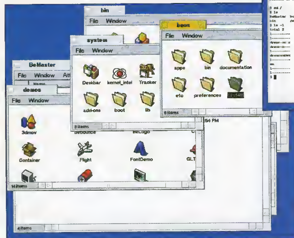
Les extensions du système : /system/add-ons

Ces extensions concernent en grande partie le support de périphériques (imprimantes, cartes sons, accélération graphique, etc.), mais aussi les convertisseurs (translators) supplémentaires, comme Bmp, Jpeg, etc.

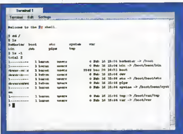
Les serveurs système : /system/servers

*BeOS* fonctionne sur le modèle client/serveur : les clients sont les applications et les serveurs, des fonctionnalités du système comme le support audio, les communications, etc. Les serveurs systèmes standards résident dans ce répertoire.

Les bibliothèques système : /system/lib




*Lassé par les écrans encombrés ? ...*




*... Préférez la solution Shell !*

Les programmes sous *BeOS* sont développés avec l'aide de bibliothèques de programmation en C++. Le code objet de ces bibliothèques est partagé par toutes les applications et prend place dans /system/lib.

- Les enregistrements systèmes : /var (/boot/var)

Ce répertoire contient des enregistrements de données relatives au fonctionnement du système. Il abrite par exemple le fichier de swap (lequel fait croire que votre machine a énormément de mémoire, en utilisant le disque).

Le répertoire /var/log/ contient en outre des données de journalisation, ayant trait à certains programmes systèmes. Par exemple, le fichier syslog enregistre tous les événements sortant un peu de l'ordinaire (comme les dérapages de la machine). Faites une expérience : tapez less syslog dans un Shell et entrez la combinaison Shift-F. Puis, plantez une de vos applications (par exemple, essayez de lancer kernel_intel, dans /system) ; le log vous indiquera alors que cette activation n'est pas possible.

Les fichiers temporaires : /tmp (lien sur /var/tmp et donc /boot/var/tmp)
Tous les fichiers temporaires créés par les applications (et normalement détruits lorsqu'on les quitte) cohabitent dans /tmp.

La configuration : /etc (/boot/etc.)
Tous les paramètres de configuration, comme le contenu du script de connexion Internet, se situent dans ce répertoire. Si, sous *Unix*, on édite souvent les fichiers de /etc avec un éditeur de texte, il demeure recommandé, sous *BeOS*, de configurer son système uniquement à l'aide des outils graphiques prévus à cet effet.

**Le disque principal : /BeMaster**
Ce répertoire est en fait un lien sur /boot. Les fichiers systèmes, du point de vue de l'utilisateur, résident sur :/BeMaster/beos. Nous avons vu que les répertoires situés à la racine, comme bin/ ou etc/, accueillaient les fichiers système de *BeOS*. Le répertoire que nous découvrons maintenant en possède également : il y a de quoi se perdre. En fait, /BeMaster/beos/ renferme tous les fichiers que l'utilisateur peut avoir à modifier ou utiliser.

Les programmes standards : /BeMaster /beos/apps
Ce répertoire comprend tous les programmes standards, comme le client mail *BeMail* ou le butineur Web *NetPositive*.

La documentation : /BeMaster/ beos /documentation
Contrairement à d'autres systèmes commerciaux qui ne donnent de documentation technique qu'aux personnes fortunées, capables de se payer les kits de développement, *Be* offre une documentation complète au format Html.

Les outils de configuration : /BeMaster /beos/preferences
Si la configuration système se trouve dans /etc, les outils permettant aux utilisateurs d'établir cette configuration sont situés quant à eux dans ce répertoire.

Le kit de développement : /BeMaster /develop
Vous trouverez dans ce répertoire tout ce qu'il faut pour développer des applications pour *BeOS* en C++ : les fichiers d'en-tête (dans headers/), les bibliothèques (dans lib/) et les outils, comme le débogueur bdb ou le kit gnupro dans tools/.

**Fin du voyage, retour à la maison : /BeMaster/home**
*BeOS*, contrairement aux "vrais" *Unix*, représente un système mono-utilisateur (trois fois hélas, car on a alors bien du mal à apprécier *BeOS* en famille !). Sous *Unix*, chaque utilisateur possède un répertoire personnel, situé dans home. Ici, le répertoire home symbolise votre répertoire attitré ; considérez-le comme votre racine propre. Ne modifiez jamais directement les autres répertoires : contrairement à *Unix*, il n'existe pas d'utilisateur privilégié, si bien que tout le monde peut toucher à l'intégralité du système de fichiers et commettre des erreurs irréparables. Toutefois, il s'agit d'un phénomène bien connu sous DOS ou *Windows*.

*Eric Anski*



*La configuration, c'est dans /etc.*

## Apprendre à se déplacer

*Pour bien comprendre le système de fichier de BeOS, il convient de maîtriser les quelques commandes Shell permettant de se déplacer dans le système de fichiers. Eh oui, il est sympathique de cliquer à droite et à gauche, mais on éprouve une certaine difficulté à se rappeler que telle fenêtre correspond au sous-répertoire de telle autre fenêtre, etc. Donc, arrêtez de vous plaindre et lancez un terminal (dans le répertoire apps) !*

*Dans un premier temps, il serait agréable de connaître notre position dans le système de fichier. Pour découvrir la position du Shell, il faut avoir recours à la commande pwd :*

```
$ pwd
/boot/home
```

*Nous nous trouvons donc pour l'instant dans le répertoire home, sous-répertoire de boot, qui se situe à la racine de l'arborescence (rappel : un arbre informatique n'a qu'une racine !).*

*Nous pouvons maintenant afficher la liste des répertoires de /boot/home avec ls :*

```
$ ls
Desktop            befs_txt     mail
DialUpNetworking   cd           people
SampleMedia        config       queries
```

*Ensuite, pour changer de répertoire, utilisez cd. Par exemple, allons dans le répertoire people :*

```
$ cd people
$ pwd
/boot/home/people
```

*Pour remonter d'un cran, on utilise cd... et cd / vous fait remonter jusqu'à la racine. N'hésitez pas à consulter nos articles "En pratique Unix/Linux" pour approfondir vos connaissances en matière de Shell.*

# Réseaux : le partage de fichiers sous Warp Server

*Cette chronique a exploré jusqu'ici les couches basses du réseau sous OS/2, ainsi que les extensions spécifiques au protocole TCP/IP. A présent, il est temps d'évoquer les services de haut niveau du réseau sous OS/2, à savoir la mise en place du partage de ressources.*

Cet article s'attardera plus particulièrement sur l'installation et la configuration de ce module. Avant tout, rappelons qu'*OS/2* et *Lan Server* représentèrent en 1988 l'une des premières solutions client-serveur de partage de ressources pour PC. Dès ses débuts, ce remède a intégré des fonctionnalités en standard que certains concurrents plus récents (comme *Windows NT*) ne proposent que sous forme d'options payantes. C'est le cas, par exemple, du partage de ports de communication ou de la mise en place de quotas sur les répertoires personnels.

L'installation du partage de fichier, au cours de la mise en place de *Warp Server*, se résume schématiquement à une case oui/non ; aussi peut-il être utile de revenir sur cette installation par défaut après coup. Voilà en quoi consiste l'objet du présent article.

**Un outil de configuration simple et efficace**

C'est dans le dossier «fonction de réseau fichiers et imprimantes» que l'on trouvera l'outil «installation et configuration des fonctions réseau local OS/2». Ce programme va en fait permettre de parachever l'installation du système.

Il s'ouvre sur une boîte de dialogue proposant des options aussi diverses que le création de disquettes pour l'installation d'un poste client ou la suppression pure et simple du serveur de réseau local. Seul le premier choix, installation et configuration du poste de travail, nous intéresse ici.

**Les différents types de serveur**

La boîte de dialogue suivante va nous donner moyen de préciser le rôle du serveur dans le réseau. Détaillons les trois possibilités proposées :

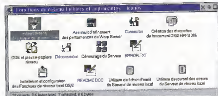
Contrôleur de domaine. Il s'agit du serveur principal du réseau local (le seul réellement indispensable). Il contrôle la base de données des utilisateurs et des ressources partagées et sert d'autorité pour l'ouverture de sessions. C'est cette configuration qui sera explorée dans le reste de l'article.

Serveur supplémentaire. Comme son nom l'indique, ce serveur vient épauler le contrôleur de domaine en matière de ressources. Pour un réseau important en termes de clients, il peut être judicieux de décharger le contrôleur de domaine et de le laisser gérer uniquement les sessions sur le domaine. Des serveurs supplémentaires doivent alors fournir des ressources fichiers et imprimante.

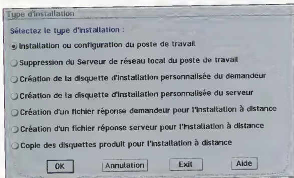
Contrôleur de domaine de Type Backup. Il s'agit d'un serveur qui vient suppléer le contrôleur de domaine en cas de panne ou d'indisponibilité. Il contient un mirroring des bases utilisateurs et ressource du contrôleur de domaine, pour répondre à sa place si celui-ci ne se trouve pas en mesure de le faire.

**Un petit passage par MPTS**

L'étape suivante concerne la configuration MPTS, laquelle permet de définir carte et protocoles. Ce module a déjà été évoqué en long et en large dans le second article de cette série ; aussi, nous ne reviendrons pas dessus, sauf pour dire que le protocole NetBios demeure indispensable pour le partage de ressources et que le système refusera d'aller plus loin si



*Le dossier de configuration du réseau local.*



*Demandez le programme !*

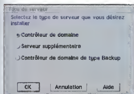
celui-ci n'est pas installé dans MPTS.

A la sortie de MPTS, si le protocole 802.2 d'IBM n'est pas mis en place, on verra s'afficher une boîte de dialogue précisant qu'en l'absence de ce protocole, le système ne pourra pas gérer les stations sans disque bootant sur la carte réseau.

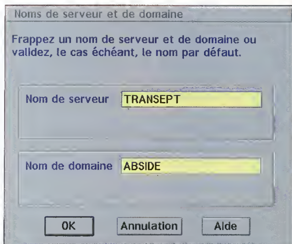
L'étape suivante nous emmène sur une boîte de dialogue proposant d'installer un nouveau module serveur, d'en configurer un déjà existant ou de valider les modifications.

**Installation d'un nouvel élément**

*Warp Server* dispose d'un certain nombre




*Le triptyque des serveurs.*




*Configuration du module serveur.*

de modules optionnels que nous ne détaillerons pas ici. On peut cependant évoquer :

- Serveur de réseau local. Comme son nom l'indique, il symbolise le coeur du serveur.

Le HPFS 386 : système de fichier 32 bits au cache paramétrable et illimité (le HPFS classique est 16 bits et ne supporte que 2 Mo de cache maximum).

La sécurité locale HPFS 386. Une fois le fameux système de fichier installé, il y a moyen d'activer sur le serveur une sécurité des fichiers en fonction de la session ouverte.

Tolérance de panne disque dur. Cet élément offre ni plus ni moins de faire du mirroring des disques du serveur sur la même machine, ou mieux, sur un serveur backup, pour conserver l'intégralité de la machine en cas de crash disque ou d'indisponibilité sérieuse du serveur.

Source de synchronisation. Il propose de synchroniser les horloges des clients avec celles du serveur à chaque connexion.

Interface graphique. Outil de gestion des utilisateurs et des ressources, que l'on détaillera dans le prochain article.

Messagerie réseau. Support d'un système de message par *popup*.

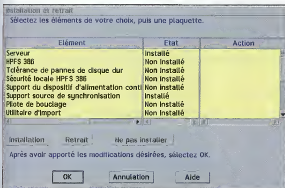
**Configuration d'un élément**

En fonction des éléments installés, le système proposera de configurer ceux-ci. La configuration la plus importante reste celle du module serveur.

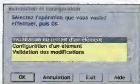
Cette configuration va permettre de définir le nom du serveur et du domaine, ainsi que les services de base lancés au cours du démarrage du serveur. Enfin, elle propose également de réinitialiser la base des utilisateurs et des ressources (cela s'avère utile, si celle-ci est corrompue ou si l'on a perdu le mot de passe administrateur).

Après la configuration des éléments, il ne reste plus qu'à valider les modifications. Le système opère alors quelques modifications dans les fichiers configuration, config.sys, ibman.ini et ibmcom.ini, avant de proposer de redémarrer le serveur. Lors du reboot, on obtient un serveur tout neuf et parfaitement configuré selon ses besoins.

*Antoine Sabot-Durand*




*Les éléments du serveur.*




*La configuration du serveur.*

# Relier une station Linux à un Palm Pilot

*Si le portage de Linux sur le Palm relève de l'anecdote, les outils de connexion avec les machines Unix sont, quant à eux, très fonctionnels.*

Largement plébiscités par la presse et les utilisateurs, les Palm Pilot de 3Com remportent haut la main la première place des ventes de PDA. Véritables ordinateurs qui tiennent dans la paume de la main, les Pilot se commandent à partir d'un stylet et reconnaissent votre écriture. Utilisables partout, ils permettent d'exécuter toutes sortes d'applications, de prendre des notes à tout moment et gèrent le planning de l'utilisateur en temps réel. L'intérêt de relier un Pilot à un ordinateur personnel est justement de récupérer ses notes et d'installer de nouvelles applications. D'office, la machine est livrée avec un socle de bureau offrant à la fois alimentation et connexion série standard. En revanche, les logiciels fournis sur le CD d'accompagnement ne tournent que sous *Windows*. Voici comment faire pour parvenir néanmoins à utiliser un Palm Pilot avec une station sous *Linux*.

**Configuration**

La première des choses à faire est de configurer le port série sur lequel vous allez connecter votre Palm. Les utilitaires *Setserial*, *Statserial*, *Kermit* et *Seyon* peuvent vous être de quelque secours si vous rencontrez des difficultés à ce niveau. A noter que le problème le plus fréquent qui puisse arriver lors de cette première étape est la non reconnaissance du port dans le Bios de votre PC. Le symptôme habituel de ce petit désagrément se traduit par un type d'Uart égal à "unknown" (Uart = universal asynchronous receiver-transmitter).

Si tout se passe bien, vous verrez apparaître des choses sensiblement équivalentes aux informations suivantes :

```
#dmesg (là où vous voyez ou non le port)
..
Serial driver version 4.13 with no seria
options enabled
tty00 at 0x03f8 (irq = 3) is a 16550A
tty01 at 0x02f8 (irq = 4) is a 16550A
...

# setserial /dev/ttyS0 (la ou l'UART est vue
ou non pour le port)
/dev/ttyS0, UART : 16550A, Port: 0x03f8, IRQ:
4
```

Si vous utilisez plus de deux ports série, et n'aimez pas jouer avec les connecteurs ou les "Switch boxes", gardez présent à l'esprit que les Irq sont entrelacés pour les ports 2 et 3, et que vous aurez à gérer votre utilisation des ressources, ou à utiliser plus d'Irq. Dans ce contexte, vous avez tout intérêt à recourir aux devices ttyS1à ttyS4, lesquels permettent un arbitrage système efficace et un accès aux ports (par le biais de fichiers de lock) en lieu et place des classiques et habituels cua1 à cua4. Si vous définissez un lien symbolique /dev/pilot, vous n'aurez qu'à vous préoccuper qu'une seule fois de cet aspect des choses. Par ailleurs, la création d'un lien vous donne la possibilité de lui attribuer des protections pour offrir l'accès

**Les commandes à avoir**

*Pilot-Xfer*
*LE programme de base, celui qui vous permet de tester votre liaison, et de charger vos premiers utilitaires (.prc et .pdb) sur votre PP.*
*Getrom*
*... Et pas Getrom.prc ni Getrom2.prc, qui sont à charger sur le PP via Pilot-Xfer. Elle sert à créer un fichier dump de la Rom du PP, pour Xcopilot.*
*Pi-Getrom*
*Copie de la Rom, sans requérir la présence d'un programme de service sur le PP.*
*Debugsh*
*Vous 'logge' sur le debug monitor du PP et vous permet de passer des commandes*
*pilot-debug : interface graphique du debugger.*
*Dplsh*
*HotSync en mode ligne.*
*install-datebook/memo/todos/user*
*Installe, à partir de la machine Linux, les données contenues dans le fichier texte dans la rubrique correspondante sur le PP.*
*Mémos*
*Récupération des mémos en format texte.*
*Read-expenses/ical/todos*
*Export des données par rubrique du PP dans un fichier texte.*
*Pilot-Addresses, Pilot-Clip*
*Utilitaires d'import-export de données d'un type spécifique (carnet d'adresses et clipboard) entre le PP et Linux.*
*Pilot-Mail*
*Client POP3 et mailer.*




*Plutôt que de vous faire des cheveux, choisissez l'option KPilot.*

au pénphérique pointé par un utilisateur non privilégié.

**Les interfaces graphiques**

Pour ceux qui n'aiment pas le mode ligne, il existe diverses interfaces graphiques pour les applications Palm Pilot sous *Linux*. La première et la plus prometteuse des interface est *KPilot*, consacrée à *Kde*. Disponible tant en format Tar qu'en Rpm, cette interface constitue une raison supplémentaire de succomber à *Kde*. A la première utilisation, l'utilisateur doit suivre le menu de configuration afin de spécifier le périphérique à adresser, le nom de l'utilisateur sur le PP, et si l'option SyncFiles doit être active Cette dernière option permet de préparer, par un drag'n drop, les fichiers chargés sur le PP lors de la prochaine HotSync. Ensuite, l'option "Local Overrides Pilot" définira qui de la machine *Linux* ou du PP possède l'information la plus récente. Le PP, plus mobile, a des chances d'être à jour ; aussi, cette option se voit invalidée par défaut.

Les autres options sont relatives au démarrage du démon de synchronisation et à celui de l'interface lorqu'une synchronisation est demandée du côté PP.

Vous pouvez également être amené à configurer des conduits. Un conduit désigne simplement une interface spécifique pour un type de données établi, comme par exemple les données de type ToDo (liste de tâches) ou pop (E-mail). La mise en service de conduits permet donc de prendre en compte de façon spécifique diverses classes de données. Il ne vous reste plus qu'à connecter votre PP, et à effectuer votre première sauvegarde. Vous verrez alors le message signalant le début de transfert, ainsi que la progression, indiquée par un bar graph dans une "Pop Up Window" séparée. Ensuite, l'environnement, très intuitif, permettra de faire des imports/exports et de modifier ou de créer depuis cette même interface des données type adresses ou mémo. En outre, *KPilot* possède des api pouvant interfacer votre PP avec d'autres applications. Au vu de la facilité d'utilisation et des possibilités ouvertes par cette interface, de nombreux outils sont à attendre dans les mois à venir.

*PilotManager* est une autre interface aux différents outils. Plus ouverte que *KPilot*, c'est également la plus ancienne, mais elle peut séduire bon nombre de personnes. A essayer absolument. Enfin, la plus récente, PUI (Pilot to Unix Interface), est à suivre de près, mais reste trop limitée et trop instable pour une utilisation au quotidien.




*Calme Pilot ?*

Nous verrons dès le mois prochain comment utiliser au quotidien un Palm Pilot avec une station *Linux*.

*Christophe Le Connelier*

# Comment programmer un jeu en 3D ?

*Oui, vous aussi vous pouvez programmer des jeux 3D, tels que Doom ou Quake. Il suffit simplement d'assimiler quelques principes mathématiques de base.*

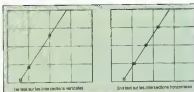


*Wolf au-dessus d'un nid de courts coups..*

Le but de cette rubrique est de vous familiariser avec les techniques et les algorithmes appliqués dans des jeux tels que *Wolfenstein*, *Doom* ou *Quake*. Pour cette toute première partie, nous nous intéresserons exclusivement au Raycasting, à savoir la technologie utilisée dans *Wolfenstein 3D* et la première à immerger le joueur dans son univers de jeu. *Doom* et *Duke Nukem* utilisent un Raycasting plus évolué, que nous étudierons ultérieurement ; néanmoins, leurs techniques découlent directement de celle de *Wolf 3D*.

**Posons le problème**

Dans un jeu comme *Wolfenstein 3D*, notre but en tant que programmeurs est triple. Nous devons afficher un sol déformé en perspective et dont le point de fuite se déplace sur l'horizon (car le joueur est mobile et son regard est tourné vers l'horizon). Ensuite, nous devons savoir à quel moment il convient d'arrêter de dessiner le sol pour passer aux murs (verticaux, eux).



*Acrné de fièvres...*

Enfin, nous devons dessiner les murs, lesquels sont aussi déformés en perspective avec un point de fuite également situé sur l'horizon (nous considérons en effet que notre joueur pourra regarder à droite et à gauche, mais pas en haut ni en bas).

**Le principe**

Commençons par simplifier : nous allons nous représenter l'univers 3D du jeu en le regardant du dessus. Eh oui, d'un point de vue aérien, un niveau de *Wolfenstein 3D* revient ni plus ni moins à un plan en 2D, sur lequel on peut observer tantôt du sol, tantôt des blocs de mur. C'est d'ailleurs parce que l'on peut les réduire à cette plus simple représentation que l'on dit de *Wolf*, *Doom* et autres *Duke Nukem* qu'ils sont en "fausse" 3D.

Pour savoir ce que regarde le joueur (donc, ce qu'il va falloir afficher sur l'écran), nous allons le symboliser par un point sur notre plan et faire partir de ce point un cône de vision qui s'étend jusqu'aux limites de ce même plan. L'angle de ce cône de vision, appelé focale ou encore FOV (Field Of View) est laissé à votre bon vouloir, mais nous vous conseillons d'opter pour du 90 degrés, ce qui correspond le mieux au champ de vision minimum d'un être humain. Pour afficher en 3D ce que voit le joueur, nous allons diviser le cône de vision en rayons (lesquels partent tous de l'utilisateur). Le principe est le suivant : chaque rayon tracé sur notre plan correspond à une bande verticale de pixels à l'écran (votre mode d'écran fait 320 pixels de large ? Alors, il faudra dessiner 320 rayons sur votre plan). Ce qui est "vu" par le rayon se retrouve automatiquement dessiné dans cette bande. Voici ce que cela donne :

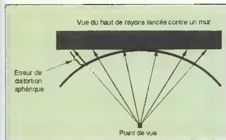
1) Chaque rayon part du joueur, donc le dessin de chaque bande verticale commencera simultanément à partir du haut et du bas de l'écran (si on considère que la ligne d'horizon est au milieu de l'écran, la perspective de la moitié supérieure de l'écran est symétrique à celle de la moitié inférieure).

2) Le rayon passe au-dessus du sol ? Alors on affichera dans la bande verticale autant de pixels correspondant à la couleur du sol (en partant du bas et en allant vers le milieu de l'écran) et du plafond (en partant du haut et en allant vers le milieu de l'écran).

3) Le rayon heurte un mur ? Alors on arrête là son parcours (a priori, le joueur ne peut pas regarder à travers le mur) ; on comble le milieu de la bande verticale par des pixels à la couleur du mur et on passe au rayon suivant.

**Un peu de rigueur**

Notre démonstration du Raycasting est très théorique. Pour pouvoir l'appliquer, il convient d'avoir un peu plus de rigueur, c'est-à-dire de la traduire en problème mathématique. Pour commencer, il faut que notre plan en 2D soit gradué, cela afin d'attribuer des coordonnées à notre joueur et de pouvoir situer les intersections entre les




*Eh oui, la distorsion, c'est son rayon.*

rayons et les murs. Mettons que notre carte tienne dans une grille de 16x16 cases, chaque case correspondant à une dalle au sol ou à un bloc de mur. Comme le joueur peut également se déplacer sur une case, nous allons encore diviser chacune d'elles en 64x64 unités afin d'obtenir des coordonnées x et y précises. Arrivé à ce point, notre joueur pourra se déplacer sur une surface de 1024x1024 unités.

C'est bien gentil de dire qu'on lance des rayons, mais encore faudrait-il savoir comment on récupère les informations qu'ils recueillent. En fait, pour que la chose soit mathématiquement plus correcte et donc plus simple à manipuler nous allons établir que notre rayon est comme un mètre que l'on déroule dans une certaine direction, depuis les pieds du joueur jusqu'au mur le plus proche. Arrivé au mur, le mètre affiche une distance d. On reporte cette distance sur notre bande verticale et on sait ainsi où il faudra mettre les pixels correspondant au sol, au plafond et au mur. Comme cette distance varie d'un rayon à l'autre, on aura bien l'impression qu'un mur s'éloigne ou se rapproche dès lors que l'on collera toutes les bandes verticales côte à côte.

**Calculons**

Calculer toutes les distances d possibles qui séparent le joueur de chaque axe vertical ou horizontal suivant un certain rayon, n'est pas très compliqué en soit. Il suffit de connaître les coordonnées x et y du joueur, la valeur de l'angle A que forme le rayon avec les axes horizontaux, l'abscisse x1 de chacun des axes verticaux et l'ordonnée y1 de chacun des axes horizontaux (ici, on parle bien entendu des axes qui apparaissent sur notre grille en 2D). On évalue alors la distance qui sépare le joueur de chaque axe vertical avec la formule suivante :

```
d - (x1-x) * cos-1 ( A )
```

Et on emploie ensuite celle-ci pour calculer la distance avec les axes horizontaux :

```
d = (y1-y) * sin-1 (A)
```

Mais tout le problème est en fait de savoir si une intersection entre le rayon et un axe se fait sur un mur ou non. Il faut donc retrouver les coordonnées du point d'intersection et regarder systématiquement sur notre plan à quoi il correspond. Pour retrouver les coordonnées du point d'intersection, nous allons nous servir de la distance et passer par Pythagore. Dans le cas d'un axe vertical, nous connaissons l'abscisse x1 du point et cherchons son ordonnée y1 (x et y sont les coordonnées du joueur) :

```
y1 = (sqrt( d^2
((x1-x)^2) )) + y
```

Dans le cas d'un axe horizontal, c'est l'inverse : nous connaissons l'ordonnée y1 et cherchons l'abscisse x1 :

```
x1 - (sqrt( d^2   ((y1-y)^2) )) + x
```

Le point de coordonnées (x1,y1) fait-il partie d'un mur ? Si la réponse est non, on recommence toute l'opération sur l'axe suivant. Si c'est oui et que l'on a débuté par les axes verticaux, alors on vérifie avant tout qu'il n'y a pas un mur plus proche du joueur sur un axe horizontal. Une fois que l'on a trouvé le mur le plus proche, on peut enfin reporter notre distance d sur la bande de pixel verticale de notre écran. Evidemment, il faudra penser à réduire systématiquement cette distance pour respecter les proportions de l'écran.

**Un effet de loupe**

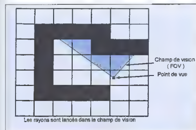
Le problème final que nous devons aborder est ce que l'on appelle la «distorsion de projection». Dans notre implémentation du Raycasting, nous utilisons des coordonnées polaires et des coordonnées rectangulaires. Ce qui a pour effet de courber les lignes de perspectives, un peu comme si notre décor était vu au travers d'une lentille sphérique. Ce problème peut être résolu en multipliant systématiquement la distance comme suit :

```
d = d * cos-1( B )
```

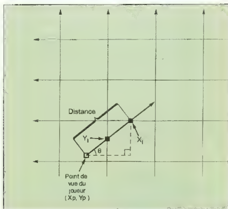
**Où B est l'angle du rayon dans la FOV**

Ce sera tout pour ce mois-ci. Au cours de la prochaine étape, nous nous attacherons à texturer notre environnement ainsi qu'à le rendre un peu plus complexe (escaliers...).

*Jérôme Lanquetot et Yann Serra*




*Labeur de carreaux.*




*Au fond des obscisses désordonnées.*

# Les principes du cryptage

*On lui attribue des vertus militaires, on prétend qu'il est le complice fidèle des agents secrets, on s'acharne à le rendre inviolable... Mais au fond, savez-vous comment fonctionne le cryptage ?*

Comment pourrait faire Roméo pour déclarer sa flamme à Juliette, sachant que le père de celle-ci ouvre systématiquement toutes les lettres qu'elle reçoit ? Roméo a trouvé : il va crypter ses messages ! Le cryptage revient à maquiller une information au point de la rendre méconnaissable. On en assure ainsi la confidentialité : un espion aura beau intercepter un message, si celui-ci est crypté, il ne pourra pas en faire grand-chose car il ne comprendra pas ce qu'il contient. L'une des premières applications historiques du cryptage est l'exemple de César. Pour communiquer avec ses centurions au nez et à la barbe de ses adversaires, l'empereur romain usait d'un fin stratagème : il écrivait toutes ses instructions en remplaçant chaque lettre par celle qui se trouvait trois positions plus loin dans l'alphabet. Ainsi, un "A" devenait un "D", un "B" devenait un "E" et ainsi de suite jusqu'au "Z" qui devenait "C" (pour faire une boucle). Par exemple, la phrase suivante :

```
DE TOUTE LA GAULE CE SONT
LES BELGES LES PLUS BRAVES
```

Etait écrite de la manière suivante :

```
GH WRXWH OD JDXOH FH VRQW
OHV EHOJHV OHV SOXV EUDYHV
```

On remarquera combien le message crypté devient sans peine totalement illisible. Bien entendu, pour pouvoir lire le message, le destinataire doit connaître la méthode pour le décrypter. Ici, il suffit de faire l'opération inverse, à savoir décaler chaque lettre de l'alphabet de trois pas dans l'autre sens. A noter que Stanley Kubrick avait usé d'un stratagème similaire pour nommer l'ordinateur fétiche de son film *2001, L'Odyssée De L'Espace* : décalez donc les lettres de "HAL" d'un pas vers la droite dans l'alphabet pour retrouver l'amusant clin d'oeil du cinéaste.

**C'est tout de même un peu simple...**
En fait, cette méthode appelée "cryptage de substitution", revient à la forme mathématique suivante :

$$f(x) = x + a$$

où f est la fonction de cryptage, f(x) le numéro dans l'alphabet de la lettre cryptée, x le numéro de la lettre à crypter et a le nombre de déplacements qu'il faut effectuer dans l'alphabet. Réduite sous cette forme, on s'aperçoit qu'il peut être très simple de compliquer un petit peu notre méthode, en rajoutant un quotient à x, par exemple. Ce qui ne serait d'ailleurs pas un mal, car le premier quidam venu serait capable, moyennant quelques

**Le carré de Vigenere**

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z | |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| O | P | Q | R | S | T | U | V | W | X | Y | Z | A | |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| P | Q | R | S | T | U | V | W | X | Y | Z | A | B | |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| Q | R | S | T | U | V | W | X | Y | Z | A | B | C | |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| R | S | T | U | V | W | X | Y | Z | A | B | C | D | |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| S | T | U | V | W | X | Y | Z | A | B | C | D | E | |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| T | U | V | W | X | Y | Z | A | B | C | D | E | F | |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| U | V | W | X | Y | Z | A | B | C | D | E | F | G | |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| V | W | X | Y | Z | A | B | C | D | E | F | G | H | |
| I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| W | X | Y | Z | A | B | C | D | E | F | G | H | I | |
| J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| X | Y | Z | A | B | C | D | E | F | G | H | I | J | |
| K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Y | Z | A | B | C | D | E | F | G | H | I | J | K | |
| L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
| Z | A | B | C | D | E | F | G | H | I | J | K | L | |
| M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| B | C | D | E | F | G | H | I | J | K | L | M | N | |
| O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| C | D | E | F | G | H | I | J | K | L | M | N | O | |
| P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| D | E | F | G | H | I | J | K | L | M | N | O | P | |
| Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | |
| R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | |
| S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | |
| T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | |
| U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| I | J | K | L | M | N | O | P | Q | R | S | T | U | |
| V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| J | K | L | M | N | O | P | Q | R | S | T | U | V | |
| W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| K | L | M | N | O | P | Q | R | S | T | U | V | W | |
| X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| L | M | N | O | P | Q | R | S | T | U | V | W | X | |
| Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| M | N | O | P | Q | R | S | T | U | V | W | X | Y | |
| Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z | |

minutes de réflexion, de décrypter les messages de César. Mais la véritable faiblesse de cette méthode tient surtout dans le fait que toutes les lettres sont transformées de la même manière. Conscient de cet état de fait, le mathématicien Vigenere a modifié quelque peu le procédé au 16e siècle. Son idée a simplement consisté à faire varier le paramètre a suivant la position des lettres dans le message. En fait, il suffit de dire que tous les a utilisés vont correspondre à autant de numéros de lettre, lesquels vont former un mot, que l'on appelle "clé" Prenons un petit exemple. Nous voulons crypter la phrase suivante :

```
FRED BOTTON IS A LOOSER
```

Et nous prenons comme clé le mot suivant :

```
DAYTONA
```

La première chose à faire est de transformer chaque lettre (que l'on appellera a(n) ) de la clé en valeur numérique :

```
04 01 25 20 15 14 01
```

Pour crypter, il ne nous reste plus qu'à faire varier dans l'alphabet la première lettre de notre phrase de a(1) positions, la seconde de a(2) positions, etc. Et lorsque l'on arrive au bout de la clé (ici, a(7) ), il suffit de recommencer avec a(1). Notre phrase ainsi cryptée devient alors :

```
JSDK QCUXPM CH O MSPRYG
```

Avantage : on ne peut même plus détecter les lettres similaires puisque celles qui le sont dans le message crypté n'ont plus aucun rapport entre elles dans le message original, et vice versa. Il existe une méthode de simple et très rapide pour crypter un message suivant la méthode de Vigenere : il s'agit du carré de Vigenere Son utilisation est la suivante : on cherche la lettre à crypter dans l'intitulé horizontal et la clé correspondante dans l'intitulé vertical. La case qui se trouve au croisement des deux colonnes représente la lettre cryptée. On constate d'emblée que le chemin inverse est bien plus difficile !

A noter que les informaticiens préfèrent employer aujourd'hui la numérotation ASCII plutôt que celle de l'alphabet. Celle-ci leur permet en effet de crypter les ponctuations ou les espaces et de faire la différence entre les minuscules et les majuscules. Pour l'anecdote, signalons que plusieurs des premiers traitements de textes sur PC/XT et AT permettaient à l'utilisateur de protéger ses documents grâce à ce principe. La clé était un mot de passe que l'utilisateur entrait une bonne fois pour toutes et qui n'était demandée qu'à l'ouverture d'un document crypté. Hélas, ce procédé avait un défaut : il suffisait à l'espion d'enregistrer un document ne contenant qu'une seule lettre, mais répétée un nombre important de fois, de regarder le contenu de ce document avec un éditeur hexadécimal et de soustraire à chaque valeur ASCII la valeur de la lettre qui avait servi au remplissage. Apparaissait alors, comme par miracle, le mot de passe de l'utilisateur.

Nous n'avons fait qu'aborder les principes basiques du cryptage. Rendez-vous au prochain numéro pour des méthodes encore plus épatantes.

*Yonn Serra*

## Le cryptage par dissimulation

*La technique du cryptage par substitution souffre d'un énorme défaut : l'espion s'aperçoit tout de suite que le message est crypté et peut d'autant plus s'acharner à trouver le moyen de lire les informations qu'il contient. De fait, les agents secrets russes pendant la Guerre Froide avaient opté pour un principe radicalement plus discret, la dissimulation. Cela consiste à noyer l'information au sein même d'un message clair. A la lecture de celui-ci, un œil non averti n'y voit que des choses anodines et ne se doute même pas qu'une information capitale vient de lui passer sous le nez. Plusieurs méthodes de dissimulation existent, allant du texte gravé en minuscules dans la pelouse d'un paysage suisse pris en photo, au morceau de musique qui, passé à l'envers, révèle l'emplacement exact des missiles occidentaux. Afin d'illustrer notre propos, voici l'exemple célèbre d'une lettre que l'écrivain George Sand (c'est une femme) avait envoyé à Alfred de Musset :*

*Je suis très émue de vous dire que j'ai bien compris l'autre soir que vous aviez toujours une envie folle de me faire danser. Je garde le souvenir de votre baiser et je voudrais bien que ce soit là une preuve que je puisse être aimée par vous. Je suis prête à vous montrer mon affection toute désintéressée et sans calcul, et si vous voulez me voir aussi vous dévoiler sans artifice mon âme toute nue, venez me faire une visite. Nous causerons en amis, franchement. Je vous prouverai que je suis la femme sincère, capable de vous offrir l'affection la plus profonde comme la plus étroite en amitié, en un mot la meilleure preuve dont vous puissiez rêver, puisque votre âme est libre. Pensez que la solitude où j'habite est bien longue, bien dure et souvent difficile. Ainsi, en y songeant, j'ai l'âme grosse. Accourez donc vite et venez me la faire oublier par l'amour où je veux me mettre.*

*Vous ne voyez rien ? Allons, relisez bien. Hé oui, la lecture de ce texte une ligne sur deux révèle une toute autre flamme...*

*http://messel.emse.fr/~gsoler/crypto1.html*

# La récursivité

*La logique d'exécution d'un programme n'est pas forcément limitée à la méthode intuitive itérative. Nous allons découvrir une finesse algorithmique : la récursivité.*

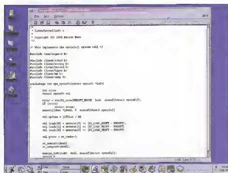
On dit qu'une fonction en C est appelée récursivement lorsqu'elle s'appelle elle-même, soit directement, soit par l'intermédiare d'une autre fonction (on parle alors de récursivité croisée). Cette méthode s'oppose à celle dite itérative. Son avantage est de fournir une plus grande clarté dans le code et de simplifier certains algorithmes. En contrepartie, sa compréhension pose toujours un petit problème pour les néophytes et, surtout, se révèle moins performante si son utilisation est non appropriée. Nous allons nous attacher à la démystifier et à démontrer sa puissance.

**Notion de récursivité**

La récursivité fait appel à un mécanisme de pile, permettant de mémoriser des résultats antérieurs. Lorsqu'une fonction s'appelle elle-même, on peut prévoir qu'au bout d'un moment (lorsqu'une condition est remplie), la dernière génération de la fonction appelée se termine par un retour, qui reviendra à l'avant-dernière génération et ainsi de suite, jusqu'à la première génération. La mémorisation de l'état de chaque génération constitue la propriété fondamentale de la récursivité. Cette mémorisation couvre en particulier les variables locales de la fonction ; le langage C leur alloue dynamiquement de la mémoire, lors de l'entrée dans la fonction et la désalloue à chaque retour.

**Intérêt de la récursivité et savoir-vivre**

La récursivité offre un grand intérêt théorique ; en effet, de nombreux algorithmes, par exemple de tri ou de recherche, s'expriment facilement sous forme récursive. S'il existe des cas où l'on a moyen d'éliminer la récursivité assez aisément, notamment pour l'algorithme de recherche dichotomique, il en va tout autrement dans d'autres cas de figure, notamment pour la raison de mémorisation. Dans ces derniers cas, autant faire confiance aux mécanismes du langage, plutôt que de gérer soi-même des structures de pile. Toutefois, cela demande un minimum de bon sens et de savoir-vivre, et le respect de quelques règles simples permet d'optimiser la partie récursive, aussi bien en termes de performance que d'allocation mémoire (ce qui reste directement lié). En langage C particulièrement, faire appel à la récursivité reste peu coûteux (appel à la fonction et allocation de mémoire rapide) et les différentes générations d'une fonction appelée récursivement ont un accès partagé aux variables externes. Cela évite de surcharger les listes de paramètres des appels récursifs avec des variables "inutiles", qui consommeraient de la mémoire sans raison. Dans le même ordre d'idées, il convient de limiter la fonction récursive à ce qui est nécessairement récursif, en y appelant non récursivement des fonctions destinées aux traitements normaux. Ainsi, on met mieux en évidence la partie récursive et on évite de surcharger la fonction en variables locales inutiles, qui seraient allouées à chaque appel récursif. Ces quelques principes permettent d'obtenir un code plus homogène, lisible et surtout optimisé.



C s'édite.

**Exemple de récursivité**

Nous allons prendre un exemple simple pour illustrer le mécanisme de la récursivité : la permutation d'un ensemble de n éléments. Voici une méthode exprimée sous forme récursive :

```
permuter(i) :
si i = n alors afficher la permutation
sinon pour j variant de i à n
{
        échanger les éléments i et j
        permuter(i+1)
        échanger les éléments i et j
}
```

Elle consiste pour une position i à déposer là successivement tous les éléments j de i à n qui suivent i et à appeler récursivement la permutation pour la position suivante. Le second échange vise à remettre à sa place l'élément j avant de prendre le suivant. Même dans un cas aussi trivial que celui-ci, il n'est pas toujours aisé de comprendre le mécanisme de la récursivité ; illustrons-le avec un petit exemple concret. Considérons une liste de trois éléments (0, 1 et 2) et voyons comment fonctionne notre algorithme. Nous prenons la convention du langage C en attribuant au premier élément l'index 0 (ainsi, la condition d'arrêt n'est plus pour i = n, mais pour i = n-1).

```
0    1    2
```

Nous rentrons dans la fonction avec i=0 et n=3 ; j va varier de 0 à 3 exclus. Nous échangeons donc les éléments i et j, qui sont le même premier élément (0). Nous appelons alors la fonction récursivement avec un nouveau i, qui vaut 1. Nous intervertissons i=1 avec j=1 et recommençons la récursivité pour arriver sur la condition d'arrêt i = (n-1), puisque nous avons commencé à 0. Nous affichons donc :

```
0    1    2
```

Nous sommes arrivés à la dernière génération et remontons donc d'une classe. Nous faisons le deuxième échange de la boucle, avec i = j = 1. Puis, nous incrémentons j et échangeons les éléments i et j (1 et 2). On recommence l'appel récursif avec i=2. Nous rencontrons donc la condition d'arrêt et affichons la liste :

```
0    2    1
```

Et ainsi de suite... Tant que j n'a pas atteint la valeur n pour la première génération, on continue le processus et on obtient finalement un affichage du type :

```
0 1 2
0 2 1
1 0 2
1 2 0
2 1 0
2 0 1
```

Voici un exemple de programme mettant en œuvre cet algorithme. Grand avantage du C, ce programme se compile sous n'importe quel environnement (du moment que le compilateur est ANSI C).

```
#include <stdio.h>
#include <malloc.h>
/* Prototypage de fonction */
void permute(int);
void affiche();
void swap(int,int);
/* Variables globales */
int *T;
int N;
void affiche()
{
        int   j;
        for (j=0 , j<N , j++)
                printf("%2d",T[j]);
        printf("\n");
}
void swap(int i, int j)
{
        int k,
        k = T[i];
        T[i] = T[j];
        T[j] = k;
}
void permute(int i)
{
        int j;
        if (i == N-1)
                affiche();
        else
                for (j=i ; j<N , j++)
                {
                        swap(i,j);
                        permute(i+1);
                        swap(i,j);
                }
}
void main()
{
        int i,
        printf("Entrez le nombre d'éléments : ");
        scanf("%d",&N );
        if ( !(T = (int *)malloc(N*sizeof(int))) )
        {
                printf("Malloc impossible\n" );
                exit( 1);
        }
                for (i=0 ; i<N ; i++)
                T[i] = i;
        permute(0);
}
```

Il n'y a rien de bien compliqué dans ce programme. Nous avons volontairement utilisé un pointeur sur entier pour le tableau des éléments, afin de nous rappeler les articles précédents. Evidemment, pour cet exemple, on pourrait se contenter d'un tableau de dimension fixe.

**Récursivité contre itérativité (et autres barbarismes)**

Réaliser le même programme sans employer la récursivité n'est pas impossible, mais se montre plus délicat. Il faut s'appuyer sur un tableau qui mémorise pour chaque position i celle qui a permuté avec la position i (ce qui correspond à j dans notre exemple). Ce tableau de position – appelons-le P contient des variables P[i] variant de i à N-1. Lorsque cette variable est égale à N-1, la boucle des permutations pour une position i est achevée et on recule alors d'une position. Autrement, on augmente P[i] de 1 et on avance jusqu'à la fin du tableau en faisant P[i] = i pour toutes ces positions. La fonction itérative correspondante est la suivante (pour simplifier, on considère des tableaux de dimension fixe) :

```
int T[10],
int N;
int P[10];
void permut()
{
        int i=0;
do      {
        for ( ; i<N ;
i++)
                P[i] =
i;
        affiche();
        for (i=i-2 ;
i>=0 && P[i] == N-1 ;
i--)

swap(i,P[i]);
        if (i >= 0)
        {
                swap(i P[i]);
                P[i]++;
                swap(i P[i]);
                i++;
        }
} while (i> 0);
}
```

Cette variante itérative est bien plus lourde à mettre en place que son homologue récursive. Cependant, elle se révèle plus rapide, mais l'écart en performances n'est que rarement significatif. Mesurer précisément la différence en temps d'exécution des deux méthodes ne s'avère pas très évident, mais on peut tabler sur environ 25 % d'écart. Cependant, il ne faut pas se focaliser sur ces différences de vitesse de traitement ; dans la majorité des cas, le gain de temps de développement et de débogage ainsi que la clarté du code constituent des éléments aussi (voire plus) importants. La récursivité représente vraiment un élément fondamental et pour peu que vous ayez à gérer des listes chaînées et à parcourir des arbres, cette méthode reste vraiment la seule satisfaisante, d'autant que la plupart des algorithmes l'utilisent.

*Stéphone Grosjeon*

# Les objets répartis en Java

*Nous allons voir ce mois-ci comment créer facilement des applications client/serveur en Java, à l'aide de Remote Method Invocacion.*

Le RMI, littéralement "appel de méthodes à distance", désigne un mécanisme grâce auquel le programmeur peut répartir son application sur un réseau. Dans un programme Java normal, chaque thread qui s'exécute appelle des méthodes d'objets, en leur transmettant des arguments et en récupérant le résultat. Avec RMI, ce schéma s'applique sur tout le réseau. Un thread s'exécutant sur une machine peut appeler les méthodes d'un objet résidant sur une autre machine, de manière quasiment transparente. Plus concrètement, nous avons ici deux types de programmes : d'une part, les "serveurs", qui créent des objets et les mettent à disposition du réseau, et d'autre part les "clients", qui appellent à distance les méthodes des objets exportés par les serveurs. Prenons un exemple pratique : un objet *Windows2000* possède une méthode "click" qui prend en paramètre le bouton sur lequel l'utilisateur a cliqué et renvoie le code de l'erreur de protection générale qui est apparue. Dans un programme normal, on écrirait :

```
erreur=windows2000.click(démarrer);
```

Lorsqu'on utilise RMI, cette ligne de code reste identique, mais prend une signification radicalement différente, puisqu'elle déclenche toute une série d'opérations. En premier lieu, le paramètre attribué à la méthode (ici, le bouton démarrer) se voit sérialisé, afin de pouvoir être transmis sur le réseau. Puis, un message est envoyé au serveur, qui transporte ce paramètre sérialisé et lui demande d'appeler la méthode "click" sur l'objet *Windows2000*. A la réception de ce message, le serveur désérialise le bouton démarrer, avant d'effectuer l'opération sollicitée avec ce paramètre. Une fois l'exécution de la méthode "click" terminée, le serveur sérialise le résultat et le renvoie au client ; celui-ci le désérialise et rend la main au thread qui a déclenché l'appel Tout un protocole est donc nécessaire pour faire à distance ce qu'on effectue habituellement en local, mais heureusement, grâce aux outils fournis par Java, le programmeur n'a pas à gérer tout cela lui-même. Il n'a (presque) rien à faire pour que ça marche. Supposons à présent qu'on soit réellement en train de programmer cet exemple et suivons les différentes étapes.

**La programmation avec RMI**

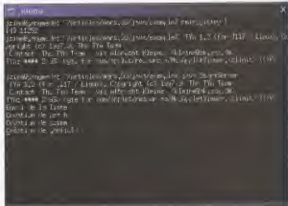
Pour rendre un objet accessible via RMI, la première chose à faire consiste à spécifier la(es) méthode(s) que l'on sera en mesure d'appeler à distance, et à définir celle(s)-ci dans une interface. Celle-ci doit hériter de Remote, pour signaler qu'il s'agit d'une interface exportée sur le réseau. Dans l'exemple sus-cité, on aurait ainsi l'interface *Windows2000*, définie dans un fichier Windows2000.java :

```
import java rmi.*,
public interface Windows2000 extends Remote {
        int click(Button bouton) throws
RemoteException;
}
```

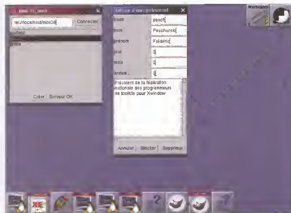
Toutes les méthodes appelables à distance doivent déceler des erreurs en levant l'exception RemoteException (ou une exception

## Les squelettes et les souches

*Le rôle de ces classes est d'assurer la gestion du protocole RMI. Lorsqu'un client accède via RMI à un objet distant, il possède en fait chez lui une instance de la classe "souche" (stub en anglais), qui possède exactement la même Interface. L'appel Naming.lookup() crée justement cette souche. Ainsi, lorsque le client appelle ensuite win2000.click(), il invoque en réalité la méthode "click" de la souche, qui sérialise les paramètres et envoie le message, puis attend la réponse, désérialise le résultat et le rend à l'appelant, qui croit ainsi manipuler le "vrai" objet Win2000 ! Quant au "squelette", san rôle est symétrique en ce qui concerne le serveur : il reçoit les messages, invoque la "vraie" méthode et retourne la réponse.*



*Une application à la "Java comme je te pousse".*



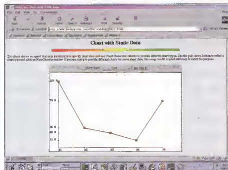
*Linux s'illustre.*

dérivée). Cette exception peut également être signalée par la machine virtuelle Java si, par exemple, on ne parvient pas à établir la connexion avec le serveur. Il demeure donc obligatoire de déclarer cette exception pour toutes les méthodes de l'interface exportée. En second lieu, on doit créer la classe qui implémente cette interface. Il faut que cette classe dérive de la classe prédéfinie UnicastRemoteObject :

```
import Windows2000;
import java.rmi.*;
import java rmi.server.*;
public class ImplWin2000 extends
UnicastRemoteObject implements Windows2000 {
        public ImplWin2000() throws
RemoteException {
                ...
        }
        public int click(Button button) throws
RemoteException {

                return err;
        }
}
```

On implémente donc toutes les méthodes de l'interface, sans oublier un constructeur pour cette classe (qui doit, lui aussi, déclarer RemoteException). On arrive maintenant au plus important. Le JDK offre en standard un outil "magique", le Rmic (pour RMI Compiler), qui permet de compiler une classe dans le but d'une utilisation par RMI. Après avoir saisi la classe ci-dessus dans un fichier ImplWin2000.java, on appelle ce compilateur par Rmic ImplWin2000. Sauf erreur, Rmic compile notre classe comme le ferait le Javac standard, mais il crée également deux classes supplémentaires, un "squelette" et une "souche" qui seront utilisées en interne par RMI sans que l'on n'ait à s'en occuper (pour les curieux, l'encadré vous donne une idée de ce qu'effectuent ces classes).



*Chosse à courbe.*

**Démarrage du serveur et connexion**

Pour faire fonctionner notre application répartie, il faut prendre en compte un dernier élément : la connexion entre le serveur et les clients. C'est là qu'intervient la seconde baguette magique : le programme rmiregistry, grâce auquel les clients peuvent contacter les serveurs. Ce programme doit tourner en tâche de fond (par exemple, sous *Linux*, tapez rmiregistry & dans un xterm pour le lancer), puis chaque serveur a l'obligation de se déclarer à lui. On écrit alors un petit programme d'initialisation en Java :

```
/ fichier InitServeur.java
import ImplWin2000;
import java.rmi.*;
class InitServeur {
        static public void main(String[] args)
{
                try {
                                ImplWin2000
win2000=new ImplWin2000();

Naming.rebind("rmi://ganymede.rubs.fr/win2000
   win2000);
                }
                catch(Exception e) {
                        // erreur
                }
        }
}
```

L'appel Naming.rebind() permet de déclarer notre nouvelle instance de ImplWin2000 auprès de rmiregistry. Le premier paramètre correspond à l'URL ; cette instance lui sera accessible par le réseau. Voyons enfin comment un client peut se connecter à ce serveur. Il doit d'abord utiliser l'appel Naming.lookup() pour récupérer une référence sur l'objet distant, puis il a moyen d'appeler les méthodes définies dans l'interface exportée (*Windows2000*, dans notre cas) ;

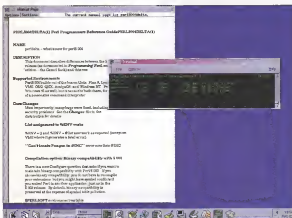
```
Windows2000 win2000;
try {
        win2000=(Windows2000)
Naming.lookup("rmi://ganymede.rubs.fr/win2000
");
}
catch(Exception e) {
        // erreur
}
```

Dès lors, le code du client se trouve en mesure de travailler avec l'objet *Win2000* de la façon habituelle, et d'appeler par exemple Win2000.click(démarrer). Toute la mécanique décrite au début de l'article sera mise en œuvre automatiquement.

*Jakub Zimmermann*

# Initiation à Perl (10) Programmation orientée objet en Perl (2/2)

*Progressons, lentement mais sûrement, dans la programmation objet en Perl.*



*En Perl de vifs tests.*

### La liaison à l'exécution

Dans la première partie de cet article, nous avions vu que, grâce à la fonction "bless", la classe à laquelle appartient un objet permet de marquer celui-ci. De cette façon, lorsqu'on appelle une fonction sur cet objet, *Perl* sait à quelle classe il se rattache et peut donc retrouver le code adéquat, grâce au mécanisme que l'on appelle "liaison à l'exécution". Il est applicable à toute fonction, et notamment à la fonction "new". Reprenons notre exemple des voitures, et imaginons que nous devions charger des données à partir d'un fichier texte. Celui-ci comprend plusieurs champs (type de voiture, marque et modèle), séparés par le caractère point-virgule. Un développeur C++ ou Java aurait tendance à écrire :

```
$ligne = <FICHIER>;
($type, $marque, $modele, $moteur) = split
(/;/, $ligne);
if ($type eq «VoitureMonoMotorisation») {
	$voiture = new
VoitureMonoMotorisation($marque, $modele,
$moteur);
} else {
	$voiture = new
VoitureBiMotorisation($marque, $modele,
$moteur);
}
```

Bien que ce code soit correct, il posera des problèmes de maintenance, dès lors que l'on souhaitera introduire de nouveaux types de voitures ; il faudra ajouter d'autres tests de conditions, etc. Le mécanisme de liaison à l'exécution introduit plus haut nous permet d'écrire un code valable une fois pour toutes, et qui se révèle plus succinct que le premier :

```
$ligne = <FICHIER>;
($type, $marque, $modele, $moteur) = split
(/;/, $ligne);
$voiture = $type->new($marque, $modele,
$moteur);
```

Attention, cependant : on ne pourra utiliser la liaison à l'exécution que si l'on a bien pris la précaution, pour chaque fonction de classe, de considérer que le premier paramètre est un nom de module (ou de classe).

### L'héritage

Dans nos exemples, nous avons différencié totalement nos deux types de voiture, mais il demeure pratiquement certain que quelques comportements seront identiques (par exemple, démarrer, accélérer, ou freiner). Pour des raisons de temps de développement et de coût de maintenance, il demeure totalement exclu d'écrire deux fois des codes semblables. Dans chaque module, il existe un tableau nommé @ISA, dans lequel nous pouvons ajouter d'autres noms de modules pour lesquels nous souhaitons utiliser les fonctions qui y sont définies. De cette manière, nous avons le moyen de créer un module "Voiture" dans lequel nous écrirons les procédures communes à nos deux types de voitures.

```
package Voiture;
sub demarrer {
	# ici le code de la méthode
}
```

Et nous préciserons donc, au début de chacun de nos deux modules "VoitureMonoMotorisation" et "VoitureBiMotorisation" qu'ils héritent du comportement de la classe "Voiture" :

```
package VoitureMonoMotorisation,
@ISA = «Voiture»;
```

et

```
package VoitureBiMotorisation;
@ISA = «Voiture»;
```

Ainsi, quel que soit le type de voiture, on peut leur appliquer la fonction "démarrer". Comme celle-ci n'est pas définie au sein du module en question, *Perl* la cherchera dans les modules désignés dans le tableau @ISA et la trouvera donc dans le module "Voiture".

Toutefois, rien ne nous empêche de définir une fonction "demarrer" spécifique au module "VoitureMonoMotorisation". Exemple :

```
package VoitureMonoMotorisation;
@ISA = «Voiture»;
sub demarrer {
```

```
        my($obj) = @_;
        print «Utilisation du starter\n» if
($obj->{'moteur'} eq «essence»);
        $obj->SUPER::demarrer() ;
        }
```

Ce mécanisme s'appelle "surcharger une fonction". "SUPER" est une référence un peu particulière qui correspond à la classe "mère" de la classe actuelle.

**Le module UNIVERSAL**

Tous les modules, que ce soient ceux livrés en standard avec *Perl* ou ceux que vous créerez, héritent implicitement du module UNIVERSAL et de trois fonctions. La première, isa (nom de module), renvoie "vrai" si le module concerné hérite, directement ou indirectement, du module précisé en paramètre. La deuxième, can (nom de fonction), renvoie "vrai" si le module, ou l'un de ses modules de base, contient une fonction du même nom que le premier paramètre. La dernière, VERSION (version requise), renvoie "vrai" si la valeur définie dans la variable $VERSION du module est supérieure ou égale à celle précisée en paramètre.

**Recherche de méthodes**

Nous avons vu que *Perl*, s'il ne trouve pas une fonction demandée dans le module courant, va ensuite la chercher dans les modules contenus dans le tableau @ISA (et récursivement dans les tableaux @ISA des classes de base), avant de faire appel à la fonction AUTOLOAD (voir l'article *Perl* et la modularité). Prenons un exemple :

```
$voiture = new
VoitureMonoMotorisation(«Renault», «R19»,
«essence»);
$voiture->freiner();
```

La recherche de la fonction "freiner" s'effectuera dans l'ordre suivant :

```
VoitureMonoMotorisation::freiner
Voiture::freiner
(les classes de base de Voiture,
récursivement)::freiner
UNIVERSAL::freiner
VoitureMonoMotorisation::AUTOLOAD
Voiture::AUTOLOAD
(les classes de base de Voiture,
récursivement)::AUTOLOAD
UNIVERSAL::AUTOLOAD
```

Si la recherche n'aboutit pas, *Perl* retourne alors un message d'erreur.

**Destruction d'objets**

Quand une structure de données n'est plus référencée et que l'on ne peut plus, par conséquent, la manipuler, *Perl* tente alors d'appeler une fonction nommée "DESTROY" juste avant de détruire l'objet en question. Il la cherche en premier lieu dans le module correspondant, et à défaut, dans les modules hérités. Une telle fonction est chargée de libérer l'ensemble des ressources qui étaient allouées par la structure de données. Bien entendu, *Perl* se charge lui-même de toutes les allocations mémoires, mais l'utilisateur a loisir d'exploiter une telle fonction pour, par exemple, fermer une connexion par socket ou enregistrer une opération dans un log.

**Accesseurs**

En programmation orientée objet, on n'aime pas trop que des fonctions extérieures au module, qui définit le comportement d'un objet, aient accès directement, en lecture ou en écriture, aux attributs de l'objet. En premier lieu, on veut être sûr que l'objet garde un état cohérent, et ce à n'importe quel moment de sa vie, afin d'éviter que ne se produisent des comportements non voulus. D'autre part, on désire assurer une évolution du comportement des objets, pour une maintenance nulle de la part des fonctions utilisant ces objets. La solution s'impose : nous allons créer des méthodes d'accès ou accesseurs, pour chacun des attributs de l'objet. Par exemple, il nous faudra une méthode pour lire l'attribut «marque» d'une voiture, et une autre pour le modifier :

```
$marque = $voiture->lire_marque();
$voiture >ecrire_marque(«Citroën»);
```

Grâce à une petite ruse de programmation, ces deux fonctions peuvent se résumer en une seule :

```
package Voiture;
sub marque {
        # récupérer la référence sur l'objet
        my ($obj) = shift;
        # y a-t-il un autre paramètre ?
        @_ ?
# si oui, opération d'écriture
$obj->{'marque'} = shift :
# sinon, opération de lecture
$obj->{'marque'};
}
```

Voici résumés en quelques points les avantages apportés par l'utilisation systématique des accesseurs : gestion des attributs calculés de la même manière que les attributs "statiques", déclenchement d'opérations annexes inhérentes à la modification d'un attribut (exemples : mise à jour de l'affichage en cas de changement de couleur, recalcul d'un autre attribut dont la valeur est devenue obsolète, etc.), accès protégé ou restreint de certains attributs. Vous voilà maintenant fin prêt pour développer vos propres objets, de la manière la plus propre qu'il soit, afin de leur garantir une pérennité maximale.

*Vincent Oneto vodko@soturne.org*

# A la découverte du Lisp

*Reconnaissable à ses multiples parenthèses, Lisp se révèle très simple lorsqu'on oublie les langages de programmation traditionnels comme le C.*

Développé en 1959 par John McCarthy au MIT, Lisp est un des plus anciens langages de programmation qui soit encore utilisé. Il repose sur un modèle mathématique : la théorie du Lambda Calcul. Le Lambda Calcul est un langage logique introduit par A. Church vers 1930 pour formaliser la notion très générale de fonction. Adapté aux traitements linguistiques et à la modélisation du raisonnement, ce langage fonctionnel est très apprécié en intelligence artificielle, notamment pour l'implémentation de systèmes experts. On le retrouve également dans le moteur du jeu *Abuse*, sans oublier l'éditeur de texte Emacs, développé en Lisp, ou bien encore *Autocad*, logiciel de CAO qui utilise une version de Lisp comme langage de programmation.

**Lisp oui, mais lequel ?**

Il n'existe pas une seule version de Lisp, mais plusieurs dialectes qui ont chacun développé leurs petites différences : MacLisp, Le_lisp de l'Inria, Vlisp, Interlisp, LM le lisp musical... Et bien s0r, Elisp, sur lequel repose Emacs. La liste pourrait être longue, surtout si on inclut les dérivés directs du Lisp comme le Scheme ou Caml. Cependant, après la création d'un comité de normalisation en 1984, la norme Common Lisp a été établie en octobre 1989. Le Common Lisp forme ainsi le noyau standard de ce langage.

## Le sens de car et cdr

*Les noms de ces deux fonctions ont un sens historique. La première implémentation de Lisp s'est faite sur un IBM-704. Sur cette machine, un mot était séparé en deux : une partie adresse et une partie décrément. Le Contenu de la partie Adresse du Registre donnait le car d'une liste et le Contenu de la partie Décrément du Registre donnait le cdr de la liste.*

C'est pourquoi ce sera d'ailleurs le langage utilisé par la suite.

**Du Lisp en Lisp**

La majeure partie du Lisp est écrit en Lisp. Il s'agit d'une de ses particularités. En effet, mis à part quelques primitives écrites en C, le reste du langage est écrit en Lisp. D'ailleurs, l'interpréteur Lisp considère toutes les fonctions que l'on peut par la suite implémenter pour un programme de la même façon que la majorité des fonctions dites standards. Ce langage, en plus d'être extensible à souhait, est interactif. En effet, il est constitué d'un interpréteur qui a pour principe de fonctionnement la boucle "Read-Eval-Print". Ainsi, l'interpréteur attend une expression qu'il va lire (read), puis il va l'évaluer (eval) et renvoyer systématiquement une valeur qui s'affichera (print). Ensuite, il reprend son cycle et attend la prochaine expression à évaluer.

**La gestion des listes**

Lisp est un acronyme pour LISt Processor. Ainsi, Lisp traite uniquement des listes et manipule des données symboliques, donc il n'y a pas de déclaration de variables ou de type de variables à faire. Une liste est une chose mise entre parenthèses, comme par exemple, (il neige). Le plus petit élément que l'on puisse trouver s'appelle un atome : 1 est un atome, tout comme "neige". En revanche, (n e i g e) est une liste de cinq atomes. De même, (1 2 il neige) forme une liste composée de quatre atomes, mais (1 2 (il neige)) est une liste composée de 2 atomes 1 et 2 et d'une liste (il neige) appelée aussi sous-liste. () désigne une liste vide, encore notée nil. nil et () ont la même valeur, à ceci près que () est une liste tandis que nil est un atome. Il se révèle d'ailleurs un petit peu particulier, puisqu'il correspond également à la valeur logique "FAUX", tout comme t (pour true) représente la valeur logique "VRAI"

**Les fonctions de base**

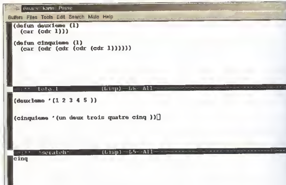
Il existe également, et heureusement d'ailleurs, d'autres listes particulières : les

## Travailler avec Emacs

*Programmer en Lisp en utilisant Emacs se révèle très simple. Après avoir activé l'éditeur, il suffit de lancer l'interpréteur Lisp par la commande :*

```
M-x lisp mode
```

*Lorsque l'interpréteur Lisp est actif, le message *lisp* apparaît en bas du buffer. Pour évaluer une expression, il vous suffira alors de taper la commande "C-x C-e" à la fin de l'expression et le résultat apparaîtra dans la ligne du bas de l'éditeur. Vous pouvez également enregistrer vos programmes dans des fichiers et travailler avec Emacs dans deux fenêtres séparées, avec d'un côté votre fichier et de l'autre, le buffer de l'interpréteur Lisp.*




*En cas de problème, Lisp tique.*




*L'affaire est triste, mais j'ai lu tous les Lisp.*

expressions symboliques. Celles-ci servent à appeler des fonctions qui vont manipuler .. des listes. Ces expressions sont de la forme (fonction argument1 argument2 ...). Un appel de fonction suit le principe de la boucle Read-Eval-Print et ramène donc toujours une valeur. Il peut être indispensable de ne pas évaluer l'expression passée à l'interpréteur, notamment lorsque l'on veut simplement la transmettre telle quelle à une fonction. Il existe une fonction qui permet d'obtenir cela : la fonction quote. Bien qu'elle ait l'air de ne servir à rien, elle est tellement utilisée qu'il en existe une abréviation : '. On y fait appel pour définir l'objet sur lequel on va travailler.

Voici un petit exemple :

```
(quote  toto) -> toto
  africa ->africa
```

Les fonctions car et cdr quant à elles servent à accéder aux différents objets d'une liste. Car ramène le premier élément de la liste passée en argument et cdr ramène la liste sans le premier élément.

```
(car  (a b c)) -> a
(cdr  (a b c)) -> (b c)
```

Essayez de taper (car (a b c)) : un message d'erreur apparaîtra, ce qui explique l'utilité de la fonction quote. Mais comment accéder à présent au second élément de la liste ? Les fonctions car et cdr sont suffisantes pour effectuer toutes sortes de combinaisons possibles. Le deuxième élément d'une liste s'obtiendra par :

```
(car (cdr  (a b c))) -> b
```

Le troisième élément sera retourné par :

```
(car (cdr (cdr  (a b c)))) = (car (cdr  (b c))) - (car  (c)) -> c
```

Ces deux compositions sont d'ailleurs suffisamment employées pour que certains Lisp en acceptent les abréviations suivantes : cadr pour (car (cdr liste)), autrement dit en langage "lispien" pour le car du cdr et caddr pour le troisième élément d'une liste, donc pour le car du cdr. On peut également conjuguer les fonctions car et cdr comme on le souhaite. Voici quelques exemples à méditer :

```
(car (car  ((1 2) a b c))) = (car  (1 2)) -> 1
(car  (cdr   (a b c))) > cdr
(cdr  (((a b c d)) 1 2 )) -> (1 2)
```

*Karine Mardal*

## Quelques livres et adresses

*Plusieurs livres peuvent aider à approfondir ses connoissances sur ce langage.*
*H. Wertz, (Common) Lisp : Une Introduction A La Programmation. Ed. Masson, Paris 1989.*
*D.P. Friedman, M. Felleisen, Le Petit LISPien. Ed. Masson, Paris 1991.*
*C. Queinnec, Lisp Mode d'emploi. Ed. Eyrolles, Paris 1984.*
*Des adresses sympatiques :*
*deux cours complets on line*
*http://surf.cs.univ-paris8.fr/vod/lisp/lisp.htm*
*http://www-ic2.univ-lemans.fr/~lehuen/cours_lisp/.*
*découvrir le Lisp musical http://www.etu.info.unicoen.fr/~fbilhaut/musicall/lisp.html.*
*apprendre à se servir d'Emacs*
*http://liawww.epfl.ch/~frei/Cours-IA/Introduction/emacs-lisp.htm*

```
M-x lisp-mode
```

*Lorsque l'interpréteur Lisp est actif, le message *lisp* apparaît en bas du buffer. Pour évaluer une expression, il vous suffira alors de taper la commande "C-x C-e" à la fin de l'expression et le résultat apparaîtra dans la ligne du bas de l'éditeur. Vous pouvez également enregistrer vos programmes dans des fichiers et travailler avec Emacs dans deux fenêtres séparées avec d'un côté, votre fichier, et de l'autre, le buffer de l'interpréteur Lisp.*

# Introduction à la programmation de X

*Dans l'univers des interfaces graphiques, X-Window est certainement celle qui offre le plus de possibilités. Cela se traduit par une plus grande difficulté de programmation.*

Avant la venue d'*X-Window*, un système *Unix* s'utilisait en mode texte (comme *MS-DOS*). La convivialité faisait alors cruellement défaut. Il n'y avait aucun moyen d'afficher des images, ni d'utiliser une souris pour faire du copier-coller... Bref, on avait ainsi une idée de l'horreur à l'état pur.

**Et vint le Macintosh**

Au milieu des années 80, surgit le Macintosh d'Apple. Tout le monde connaît son histoire et ce qu'il a apporté à la micro-informatique. Au Massassuchet Institute of Technology (MIT pour les intimes), on trouva l'idée excellente et des programmeurs de génie pensèrent à programmer un standard de développement d'interface graphique pour tous les systèmes *Unix*. De là est né *X-Window*.

Son principal aspect vient de l'indépendance du matériel d'affichage et des périphériques d'entrées. *X-Window* désigne en fait un protocole ; tout matériel reconnaissant celui-ci est capable d'afficher des textes et des graphiques. Le gros avantage vient du portage facile des applications. A cet égard, il n'est pas obligatoire de modifier le source d'un programme *X-Window*, car une recompilation suffit lors du passage d'une machine différente à l'autre. Ainsi, une application écrite sous Sun/solaris et respectant le protocole X sera portée sans le moindre effort sous *Linux*.

**Le principe du serveur**

Les ingénieurs du MIT ont conçu un modèle client-serveur, grâce auquel un programme client demande au serveur d'afficher des données. Le serveur X sera alors un terminal X et le client, un programme tournant sur une station distante. A cela s'ajoute la capacité de partager un serveur X entre plusieurs clients. Ainsi, on a moyen de regrouper sur un seul terminal X toutes les applications exécutées sur diverses stations (VAX, Sun Sparc, *HP 9000*...).

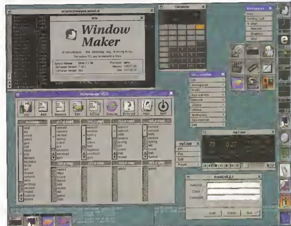
Ce modèle client-serveur utilise le protocole TCP/IP. Vous l'aurez donc deviné, la connexion client-serveur peut donc s'effectuer par le biais d'Internet. Rien ne vous empêche de lancer un programme sur une station *Unix* au Canada et d'afficher le résultat sur une machine serveur X (un PC sous *Linux*, par exemple) en France ! Toutefois, cela se traduit par une légère baisse des performances graphiques, liée à la lenteur du réseau Internet ou de la connexion. Néanmoins, sur un réseau Ethernet, cela s'avère plus facilement réalisable.

**Une interface utilisateur ? Que nenni !**

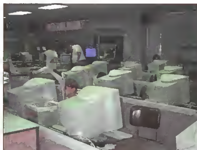
L'autre grosse différence avec les systèmes multi-fenêtrages classiques, tels que celui de l'Amiga ou du Macintosh, vient de ce que *X-Window* ne propose pas un style d'interface utilisateur. Aucun "look and feel" n'est imposé. Par exemple, une fenêtre n'est à l'origine qu'un carré sans rien autour. Par la suite, il incombe au programmeur de développer les boutons, ascenseurs et autres qui l'accompagnent. Une tâche s'acquitte de cet ouvrage : on l'appelle le *Window* Manager. *Afterstep*, fvwm, Kde sont des environnements graphiques qui proposent un *Window* Manager.

Par ailleurs, il faut bien comprendre que les boutons, ascenseurs et autres gadgets affichés par les divers *Window* Managers ne correspondent en réalité qu'à des fenêtres au sens Xlib (Xlib est le nom de la libraire principale de *X-Window*). Il en va de même pour les menus et barres d'icônes. Voilà pourquoi il devient si facile aux développeurs de créer leur propre interface utilisateur (il n'y a qu'à voir le nombre impressionnant de Window Manager existant sous *Linux*). On peut donc faire comme on veut, reprendre la GUI (Graphics User Interface) du Mac, de *Windows 95* (quoi que non, nous allons éviter...), de l'Amiga (oui, ça, c'est déjà mieux) ou même celle du Next ! Eh oui, vous avez le droit de placer un bouton qui fait "coin" en haut à gauche de votre fenêtre, avec une barre couleur arc-en-ciel et des flèches qui ne ressemblent à rien d'autre qu'à des flèches.

On aurait le droit de penser que ces différentes opérations demeurent très simples, mais hélas, mille fois hélas, tout cela cache




*Jungle à Window.*



*Au vivier de machines...*

une grande complexité. Vous devez tout faire par vous-même, car d'origine, la Xlib ne propose aucune fonction permettant d'insérer un bouton "OK" à l'endroit souhaité ; quant aux menus, ils existent encore moins ! L'écriture de sa propre interface utilisateur s'avère aussi très compliquée, car il faut interpréter les messages du serveur X (en gros, quand le serveur X reçoit un événement, souris ou clavier, il l'envoie au *Window* Manager, qui va alors exécuter l'action demandée).

Bref, le but de cette rubrique ne consiste pas à vous apprendre la méthode pour créer votre propre environnement ; il s'agit plutôt de vous aider à en comprendre le principe, tout en vous indiquant comment l'utiliser d'un point de vue développeur.

**Alors ? Comment ça marche ?**

D'abord, notre programme client demande poliment au serveur X l'autorisation de se connecter. Le serveur X va alors vérifier que l'on en a le droit, car pour chaque serveur X, on se trouve en mesure de spécifier qui peut se connecter et qui ne le peut pas. Nous nous pencherons plus tard sur ce processus, mais retenez d'ores et déjà qu'il représente une base de la sécurité de votre système *Unix*.

Une fois la connexion établie, le client a loisir de "jouer" avec n'importe quel écran du serveur X : celui-ci est apte à gérer plusieurs écrans à la fois, même si l'on contrôle rarement, en tant qu'utilisateur lambda, plusieurs écrans en simultané. Cependant, le protocole y parvient sans problèmes (attention : nous parlons d'écrans au sens physique, et non d'écrans au sens virtuel, comme le proposent certains *Window* Managers).

Pour chaque structure d'affichage (fenêtres, polices, couleurs), le serveur X alloue une ressource identifiée par un nombre unique. Ces ressources sont détruites par défaut à la fin de la connexion avec le serveur X, mais il est envisageable de modifier ce comportement. L'utilisateur peut aussi les supprimer lui-même, mais cela n'a que peu d'intérêt.

A chaque fois que notre programme client veut ouvrir une nouvelle fenêtre ou bien afficher un texte, une requête est émise vers le serveur X. Celle-ci prend place dans une file d'attente où elle sera traitée ultérieurement, car le protocole X a une nature asynchrone. L'utilisateur a l'opportunité de "forcer" le serveur à la traiter tout de suite, mais cela se traduit par une baisse des performances (les autres clients se voient alors ralentis, car leurs requêtes ne se trouvent prises en compte qu'avec du retard). De plus, monopoliser le serveur X ne signifie pas augmenter la vitesse d'affichage ; en effet, la meilleure solution reste encore de se synchroniser avec lui.

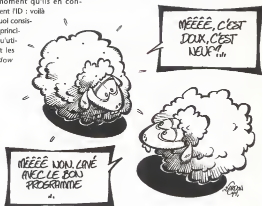
**La fenêtre est d'abord un rectangle**

L'élément de base, sous X, est la fenêtre au sens où nous l'avons défini précédemment, à savoir celui d'une zone rectangulaire. Pour créer un contour de fenêtre, nous allons utiliser d'autres fenêtres, qui constitueront alors des sous-fenêtres de la première. Afin que le client s'y retrouve, un ID est attribué à chaque fenêtre. Il peut émettre des requêtes, qui demandent de modifier leur taille, leur position ou d'y écrire du texte ou des graphismes. L'accès aux fenêtres est autorisé à tous les clients, du moment qu'ils en connaissent l'ID : voilà en quoi consiste le principe qu'utilisent les *Window* Managers. Aux ID octroyés aux fenêtres par le serveur X s'ajoute une hiérarchie de celles-ci : la plus "haute" correspond à la fenêtre racine, qui occupe tout l'écran. Elle n'est pas redimensionnable ; d'autre part, chaque écran géré par le serveur X hérite d'une fenêtre racine. Vient ensuite la fenêtre mère, qui représente en général la fenêtre principale (c'est-à-dire celle demandée par le client), puis suivent les filles (ou sous-fenêtres, qui coïcident normalement avec celles que gère le *Window* Manager). Deux fenêtres filles qui ont la même fenêtre-mère sont dites "soeurs". Afin que nous puissions nous repérer dans notre fenêtre, chacune d'entre elles bénéficie d'un système de coordonnées spécifiques : celui-ci a pour origine le coin supérieur gauche (0,0). Plus on va vers la droite, plus on augmente les x et plus on se dirige vers le bas, plus on accroît les y. Ces coordonnées ne changent pas, quelle que soit la position de la fenêtre à l'écran et ce, même si on la déplace.

Histoire de compliquer un petit peu plus les choses, toutes les fenêtres ne sont pas visibles : tant qu'on ne lui a pas adressé la requête d'affichage, le serveur X n'appelle pas les routines matérielles pour qu'elles apparaissent à l'écran. Voilà sur quel principe repose l'iconification des fenêtres : on demande au serveur X de ne plus l'afficher puis on exhibe une fenêtre plus petite : l'icône. Celle-ci se trouve la plupart du temps "en attente" d'affichage. Le moment venu, on la fait disparaître, puis on rappelle la première.

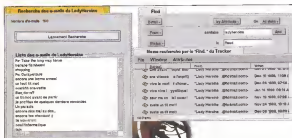
Nous nous intéresserons la prochaine fois à notre premier programme et à la gestion des événements sous X.

*Olivier Bedouet - badmax@chez.com*



# Les "queries", ou comment manipuler des fichiers

*Après avoir abordé la communication entre programmes et implémenté notre première interface graphique, terminons cette introduction à la programmation de BeOS par un aperçu sur les accès fichiers.*




*Notre application renvoie 108 E-mails, le tracker BeOS aussi (fonction «Find») : le compte est bon !*

La manipulation des fichiers par programme sous *BeOS*, peut se faire par l'entremise des fonctions écrites en C++ ou bien de fonctions POSIX (en C, donc). Nous n'allons pas trop nous attarder sur le système de fichier en lui-même ; son organisation est hiérarchique. Les données sont arrangées en arborescence de répertoires et de fichiers. La racine "root directory" symbolise le parent de tous les fichiers ou répertoires contenus dans la hiérarchie. Quand on accède par programme aux fichiers (ou répertoires), que manipule-t-on ? Pour *BeOS*, l'utilisateur se trouve confronté à des entrées (entries) et des noeuds (nodes). Une entrée correspond simplement au chemin d'accès (pathname) du fichier dans l'arborescence. Les noeuds font référence au contenu des fichiers (ou des répertoires). Dans toutes les opérations que vous effectuerez sur les fichiers, vous retrouverez systématiquement au moins l'une de ces deux notions. Nous allons rentrer dans le vif du sujet : faire de la programmation de "queries". Tandam...

**Qu'est-ce qu'une query ?**

Une query (ou requête) désigne une recherche, de fichiers, de dossiers (de tout ce que vous voulez, en fait) dans *BeOS* (accessible dans le menu File de n'importe quelle fenêtre, ou depuis la deskbar). Cette fonction va bien au-delà de ce que propose *Windows* ; en effet, la fonction de recherche donne accès à toute une combinaison de critères affinant la recherche. Exemple : je recherche tous les E-mails écrits par Ladyheroine (c'est ma copine) et que j'ai déjà lus. En langue Geek (sans la pizza dans la bouche), ça nous donne : je déclenche une recherche sur le type de fichier E-mail, en utilisant les attributs from (qui doit contenir "ladyheroine") et status (égal à "Read").

**"Je vous parle d'un temps, que les moins de 20 ans..." ("La Bohème", C. Aznavour)**

Les queries représentent les vestiges d'un temps où *BeOS* était réservé à une élite ; l'époque des Developper Release. En ces temps reculés, le système de fichier constituait rien moins qu'une base de données (API BDatabase) contenant des tables. L'accès s'y faisait grâce à ces queries. La gestion de base de données fut abandonnée du fait de son manque de vélocité au fur et à mesure que le disque se remplissait ; les queries, quant à elles, sont restées. L'interface de *BeOS* crée ce qu'on appelle des queries vivantes ("live queries") : le résultat de la recherche est actualisé en temps réel. Une fois que l'on a terminé, il est inutile de réactualiser le résultat en lançant une nouvelle recherche. Cette fonctionnalité très utile peut tout à fait être intégrée dans un programme. Nous y reviendrons. Commençons plutôt par le début. Vous croyiez en avoir fini avec notre application du mois dernier ? Eh bien, vous vous trompiez. Le code va être réutilisé, afin d'alimenter notre liste à l'aide du résultat de la requête exprimée ci-dessus. Nous allons même ajouter un compteur récapitulatif du nombre d'E-mails trouvés. En vérité, quatre parties sont à revoir dans notre application : nous supprimerons le champ de saisie. Notre bouton deviendra alors l'élément déclencheur de la recherche. Nous devons modifier la fonction AjoutDansListe() (nous n'ajoutons plus des éléments saisis, mais la liste des E-mails trouvés). Enfin, nous devons compter le nombre d'éléments récupérés et afficher ce recensement. Procédons ainsi pour la fonction AjoutDansListe() :

```
void
GuiFenetre::AjoutDansListe()
{
BEntry entry_of_query;
status_t code_retour; // nous ne sommes pas
des boeufs quand meme...
BQuery query;
 ..
...
```

Nous créons un objet de type BEntry nommé entry_of_query. Un tel objet est destiné à localiser un fichier dans l'arborescence. Il s'agit d'un chemin d'accès accompagné du nom du fichier ; par exemple, l'entrée "boot/home/gui.cpp" représente le fichier gui.cpp situé dans le répertoire "boot/home". Cet objet va nous servir à stocker le résultat de notre recherche. Nous créons l'objet vedette de cet article, query, à partir de la classe BQuery. Le reste va se dérouler en trois actes : l'initialisation de l'objet query, le lancement de la recherche et la lecture du résultat.

```
 ..
...
/* Init de la query ACTE 1 */
query.Clear(); // RAZ de la requete (on ne
sait jamais ..)
app_info info; // structure d accueil info
sur appli
be_app->GetAppInfo(&info); // Recup des infos
BVolume vol(info.ref.device); // recup du
device en utilisant la
// la structure entry_ref
```

```
query.SetVolume(&vol), // init du volume
...
...
```

Le recours à la fonction Clear() est vivement recommandé dans un cas comme celui-ci. Facultatif pour la toute première requête, il permet cependant de remettre à zéro l'objet query. Une query ne peut s'exécuter que sur un volume à la fois. Nous devons trouver ce volume. Nous passons par des voies détournées, en l'occurrence par l'initialisation d'un objet app_info (regroupant des informations concernant l'application en cours d'exécution). Il s'agit d'une structure qui pointe entre autres sur une autre structure intitulée entry_ref. Celle-ci procure un autre moyen de représenter les entrées dans la hiérarchie du système de fichier. Elle contient notamment le device (le volume donc) de l'objet auquel il fait référence. Grâce à cette caractéristique, on se trouve en mesure d'indiquer sur quel volume s'exécutera la query (SetVolume()).

```
...

query.PushAttr('MAIL:from');
query.PushString('ladyheroine');
query.PushOp(B_CONTAINS);
query.PushAttr('MAIL:status');
query.PushString('Read'),
query PushOp(B_EQ);
query.PushOp(B_AND);
...
...
```

Malgré son apparente clarté, ce code nécessite que l'on précise certains points. Ici, on construit le "prédicat" de notre requête. Les requêtes font des recherches sur les attributs des fichiers. Tous les fichiers de *BeOS* possèdent des attributs (au moins trois par défaut : le nom du fichier, sa taille et sa date de modification). Ceux-ci peuvent être de différents types (string, float, int64 le système de fichier de BeOS est sur 64 bits int32 et double). Il y a moyen de construire ses propres types de fichier et d'y ajouter tous les attributs que l'on souhaite. Le type E-mail en offre un exemple et ses attributs sont nombreux (priorité, statut, sujet, etc.).
Pour construire le prédicat, on indique l'attribut (PushAttr()), la valeur recherchée sur lui (grâce à PushString()) et enfin l'opérateur de la requête (PushOP()). Comme vous le constatez, le prédicat s'écrit à la mode polonaise. Citons quelques exemples d'opérateurs : B_EQ (égale à), B_NE (différent de), B_ENDS_WITH (se termine par).
Une question vous hante sûrement : comment connaître les attributs spécifiques de tel ou tel type de fichier ? En fait, vous n'avez aucun souci à vous faire : il suffit de lancer l'application File Types (dans les préférences *BeOS*) et d'y choisir son type de fichier. La liste des attributs doit s'afficher (voir la capture écran). En double-cliquant sur celui qui vous intéresse, vous obtiendrez son nom interne («MAIL: from» pour l'attribut From, par exemple).

```
...
char e_mail[B_FILE_NAME_LENGTH]; // Pour
recup nom des e-mails
/* lancement de la requete ACTE 2 */
code_retour = query.Fetch();
...
```

La fonction Fetch() place notre requête en arrière-plan et redonne immédiatement la main au programme.

```
...
/* recuperation du resultat ACTE 3 */
if (code_retour == B_NO_ERROR)
{
while ((code_retour =
query.GetNextEntry(&entry_of_query)) ==
B_NO_ERROR)
{
entry_of_query.GetName(e_mail); //recup nom
fichier
BListItem *UnElement = new
BStringItem(e_mail); // On    //construit un
item
ElementsListe->AddItem(UnElement); // ajout
de l'item dans la liste
}
}
...
..
```

Nous récupérons la liste des entrées trouvées, une par une, grâce à la fonction GetNextEntry(). Nous pouvons alors construire notre liste constituée du nom des fichiers trouvés (la fonction GetName() renvoie uniquement le nom du fichier contenu dans l'objet entry_of_query, qui pour sa part, contient en sus le chemin d'accès). Lorsque toutes les entrées sont trouvées, ou quand il n'y en a aucune qui satisfasse notre requête (cela peut arriver), GetNextEntry() renvoie le code retour B_ENTRY_NOT_FOUND. Pour notre query, c'est fini. Il nous faut compter le nombre d'éléments dans la liste et afficher le compte obtenu.

```
Compteur = ElementsListe->CountItems(), // on
compte le nombre d items
char buffer[32];
sprintf(buffer, 'Nombre d'e-mails . %ld',
Compteur);
Nombre_E_Mail->SetText(buffer); //
actualisation du compteur
} // Fin de AjoutDansListe()
```

Maintenant que vous êtes familiarisé avec la programmation *BeOS*, cette dernière portion de code ne devrait pas vous poser de problème. Notons seulement que : Compteur est un type int32 et que Nombre_E_Mail représente un objet dérivé de la classe BStringView, qui comme son nom l'indique, permet l'affichage de chaînes caractères. Sa construction dans GuiFenetre() occupe deux lignes de codes :

```
Nombre_E_Mail = new BStringView(BRect(10, 10,
299, 22), 'NombreMail', 'Nombre d'e-mails :
0');
top->AddChild(Nombre_E_Mail),
```

Et les queries "vivantes" ? Formulons la question en termes de fonctionnalités : nous souhaiterions mettre à jour notre liste lorsque nous supprimons un des E-mails du disque dur. L'application doit être informée automatiquement de cet événement. Les fameux Bmessages y pourvoient. Quand une query est dite "vivante", le système rend compte à l'application des événements ayant un rapport direct avec notre requête. Le message B_QUERY_UPDATE recense ces événements. La copie d'écran vous permet de remarquer que notre application renvoie 108 E-mails, dont les attributs «from» et «status» satisfont aux conditions de notre requête. En supprimant un des ces E-mails (ou si nous modifions la valeur d'un de ses deux attributs), on devrait avoir 107 E-mails. Grâce à l'utilisation de requêtes «vivantes», notre application pourra être automatiquement informée de ce changement, en recevant l'événement B_QUERY_UPDATE. Comment transformer alors notre requête moribonde en requête «vivante» ? Par une fonction : nous ajouterons ainsi query.SetTarget(this) juste avant le Fetch(). This érige notre application en destinataire des messages liés à notre query. B_QUERY_UPDATE pourra alors être intercepté et traité en conséquence dans la fonction GuiFenetre():MessageReceived(*msg) vue le mois dernier. Les queries offrent une fonction de recherche de fichiers, simplifiée mais très puissante. Le présent exemple donne un aperçu de ce que *BeOS* permet de faire en termes de gestion des données sur le disque (création de types de fichiers, association d'attributs supplémentaires, etc.). Cet article clôt notre série sur l'introduction à la programmation de *BeOS*. Pourtant, cela ne fait que commencer...

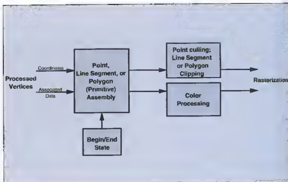
*Christophe Morvant*

# Découvrir la syntaxe d'OpenGL

*Voici le premier article d'une nouvelle rubrique consacrée à la programmation OpenGL, cette API d'accélération graphique adaptée à toutes les plates-formes.*

An'en pas douter, vous souhaitez passer à l'étape supérieure dans la conception de vos applications 3D, en leur offrant un rendu temps réel digne des performances des meilleures cartes 3D. Bien entendu, cette nouvelle série d'articles part du principe que vous savez concevoir une application 3D (un jeu...), c'est-à-dire que vous maîtrisez tous les principes mathématiques qui s'y rapportent. Ce n'est pas votre cas ? Alors, ruez-vous sur la nouvelle rubrique d'initiation à la programmation 3D, présente dans ce magazine !

Pour cette première étape, nous allons étudier la base, à savoir la syntaxe imposée par *OpenGL*. Rassurez-vous, il s'agit là d'un modèle de simplicité qui confère tout son charme à cette librairie. Les ingénieurs de Silicon Graphics Inc. (ou "SGI", le créateur d'*OpenGL*) ont en effet opté pour une méthode permettant d'appréhender de façon intuitive les diverses fonction de ce système ; elle s'appuie sur les préfixes et les "postfixes"




*Pour atteindre des sommets, on peut faire confiance à Opendjebel.*

**Les préfixes**

La méthode des préfixes est en fait un simple principe d'écriture : on rajoute les lettres "Gl" au début des noms de données, fonctions et autres constantes que l'on trouve habituellement dans le langage C, pour préciser que ces composantes se réfèrent spécifiquement à l'API et non au C générique. Ainsi, si l'on regarde les différents type de données possibles (voir tableau 1), on remarque que tous ressemblent à s'y méprendre à leurs homonymes du langage C, excepté le préfixe "Gl" : on retrouve les fameux int, dénommés "Glint", les unsigned byte rebaptisés Glubyte, etc. Certains types n'ont pas de correspondance en C comme Glclampf, lequel est un flottant défini (clamped) sur [0,1], ou bien encore Glenum, qui correspond à une valeur énumérée. La façon de faire reste la même pour les fonctions (par exemple, glShadeModel()) et les constantes, si ce n'est que celles-ci s'écrivent toujours en majuscules et ont "GL_" (GL_FLAT, par exemple) pour préfixe. Attention ! Il convient de noter un détail important au sujet de la documentation officielle d'*OpenGL* : les types des arguments ne sont pas précédés de "Gl" afin de respecter la syntaxe du C et éviter toute source de confusion.

**Les postfixes**

La catégorie des "postfixés" ne concerne en revanche que les fonctions. Il s'agit de chiffres ou de lettres que l'on place dans la fonction pour déterminer, respectivement, le nombre et le type de paramètres que l'on doit lui attribuer. Pour rendre la chose un petit peu plus claire, analysons la construction générique suivante d'une fonction *OpenGL* :

```
rtype Name{1234}{ b s i f d ub us ui }{v} (
T arg ) ;
```

"rtype" est le type de données retourné par la fonction. Il s'agit, soit de l'un de

**Les données d'OpenGL**

| *Type OpenGL* | *Nbre de bits* | *Description* |
|---|---|---|
| *Glboolean* | *1* | *booleen* |
| *Glbyte* | *8* | *octet signé* |
| *Glubyte* | *8* | *octet non signé* |
| *Glshort* | *16* | *entier court signé* |
| *Glushort* | *16* | *entier court non signé* |
| *Glint* | *32* | *entier long signé* |
| *Gluint* | *32* | *entier long non signé* |
| *Glsizei* | *32* | *entier stockant les tailles des variables* |
| *Glenum* | *32* | *valeurs énumérées* |
| *Glbitfield* | *32* | *champs de bits* |
| *Glfloat* | *32* | *réel* |
| *Glclampf* | *32* | *réel défini sur [0,1]* |
| *Gldouble* | *64* | *réel double* |
| *Glclampd* | *64* | *réel double défini sur [0,1]* |

## Dix types d'objets basiques

*Voici dans le détail la liste des objets basiques que l'on peut dessiner sous OpenGL et le nom du mode de dessin (entre parenthèses) qui leur correspond.*

● *Points (POINTS)*
*Les vertices définis entre Begin et End seront considérés comme des points indépendants, sans aucun lien entre eux.*

● *Segments (LINE_STRIP)*
*Les segments seront affichés en se servant de deux points : le ième point qui sera connecté au (i-1)ème point de la liste des vertices définis entre Begin et End.*

● *Boucles de Segments (LINE_LOOP)*
*Elles sont contrôlées de la même manière que les segments, mais contrairement à eux, le dernier point de la liste se trouve connecté au premier.*

● *Segments séparés (LINES)*
*Les segments seront affichés en prenant les points dans la liste par paires, de manière non connectée. Si le nombre de vertices est impaire, le dernier de la liste ne s'affichera pas.*

● *Polygones (POLYGON)*
*On décrit un polygone en précisant ses contours à l'aide d'une série de segments. Il ressemble donc à une boucle de segment, mais contrairement à elle, l'utilisateur peut en remplir l'intérieur suivant un état qu'il aura préalablement établi. Cependant, notons qu'OpenGL ne dessinera correctement que les polygones convexes. Par ailleurs, précisons aussi que l'ordre dans lequel les vertices se trouve défini est important pour l'ombrage et l'affichage correct des objets.*

● *Bandes de triangles (TRIANGLE_STRIP)*
*Une bande de triangles représente une série de triangles connectés entre eux (avec par conséquent deux vertices communs). Chaque sous-ensemble de trois vertices définit un triangle.*

● *Eventail (TRIANGLE_FAN)*
*Cette catégorie est identique à la précédente, sauf que les deux vertices en commun correspondent au précédent et au premier vertex de la liste (cela forme donc un éventail).*

● *Triangles séparés (TRIANGLES)*
*Il s'agit d'une sorte de mélange entre les segments séparés et les bandes de triangles ; les vertices se voient donc regroupés par trois et les triangles sont indépendants.*

● *Bandes de quadrilatères (QUAD_STRIP)*
*Leur principe ressemble énormément à celui des bandes de triangles, mais contrairement à elles, les vertices vont par quatre.*

● *Quadrilatères séparés (QUADS)*
*Le système des segments séparés, avec un regroupement par quatre des vertices, prévaut encore ici. On affiche donc d'innombrables rectangles à l'écran.*

ceux qui figurent dans le tableau 1, soit de l'un des types classiques du langage C. Le contenu des accolades correspond aux postfixes. Précisons que nous avons mis ces termes entre des accolades pour faire comprendre au lecteur que le programmeur choisirait "parmi ceux-ci" les postfixes nécessaires à sa fonction.

Le terme "T arg" désigne un argument dont on précise le type, appelé ici "T". Il correspond à un type défini par une des lettres contenues dans notre deuxième paires d'accolades (et pour savoir à quoi renvoient ces lettres, on se référera au tableau 2).

Le caractère "v", s'il est présent, indique que l'argument revêt la forme d'un tableau. S'il fait défaut, cela signifie que nous avons affaire, non pas à un seul argument, mais à une suite d'arguments. Enfin, le nombre que l'on choisira dans la première accolade représentera soit la taille du tableau, soit le nombre d'arguments.

Prenons un exemple. Lorsque l'on écrit :

```
void Normal3 {fd}( T arg ) ,
```

cela signifie que l'on peut avoir les deux déclarations suivantes :

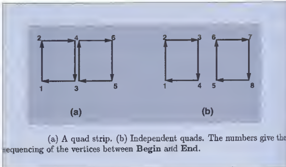
```
void Normal3f( float arg1, float arg2, float arg3 ) ;
void Normal3d( double arg1, double arg2, double arg3 ) ;
```

alors que :

```
void Normal3{fd}v( T arg );
```

peut avoir les deux interprétations suivantes :

```
void Normal3f( float arg[3] ) ;
void Normal3d( float arg[3] ) ;
```




*Carré, c'est mon produit de côtés.*

### Objets et primitives OpenGL

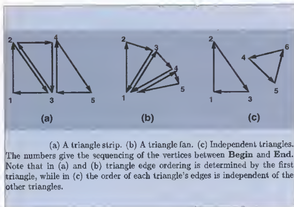
Les objets géométriques sont dessinés au moyen d'un ensemble de paramètres qui définissent les sommets de l'objets, les coordonnées de texture, les couleurs et éventuellement des normales, établies dans une paire de fonctions Begin/End. Signalons qu'il y a moyen de constituer chaque sommet (ou "vertex") grâce à deux, trois ou quatre coordonnées et qu'on peut leur adjoindre une normale, une coordonnée de texture et une couleur courantes. Tous ces attributs déterminent le comportement du sommet en fonction de la situation (ombrage, texture mapping, etc.). De cette manière, on se trouve en mesure de dessiner dix sortes d'objets basiques : les points, les segments, les boucles de segments, les segments séparés, les polygones, les bandes de triangle (Triangle Strips), les triangles en éventail (Triangle Fans), les triangles séparés, les bandes de quadrilatères et les quadrilatères séparés. Chaque sommet est défini avec deux, trois ou quatre coordonnées.

Les fonctions Begin et End tracent des objets grâce à onze valeurs énumérées, à savoir les dix modes de dessin que nous venons d'évoquer ainsi que l'état de l'objet (indiquant si un Begin/End est en cours de traitement ou non). Les commandes relatives sont donc les suivantes :

```
void Begin( enum mode );
void End( void );
```

A noter que le nombre de vertices ("vertices" est le pluriel de vertex) n'est absolument pas limité. Maintenant que nous avons abordé les notions de base, il ne nous reste plus qu'à attaquer notre premier programme dès le prochain numéro et à entamer l'étude de GLUT.

*Malek Bengougam*
*Malek.Bengougam@lemel.fr*




*L'octet aux Ram de Pythagore.*

**Signification des postfixes**

| Lettre | Type de donnée associé |
|---|---|
| b | byte |
| s | short |
| i | int |
| f | float |
| d | double |
| ub | ubyte |
| us | ushort |
| ui | uint |

# * MYGALE *

31 Bd Raimbaldi 06000 NICE
Tel/Fax 04.93.13.06.35

| PROMO DU MOIS | |
|---|---|
| Pack Deluxe | 100f |
| Quaterback 5 | 100f |

| FRAIS DE PORT | |
|---|---|
| CD/jeux/Logiciel : | 35f |
| Carte/HD/Mem | 60f |
| Tour . | 100f |
| +10f par produit en plus | |
| **10H30-12H30 / 14H-19H** | |

| AMIGA 1200 | |
|---|---|
| Processeur 68060 | 1790f |
| Copro 68882/40 Mhz PGA | 299f |
| Apollo 1230/40 1230+copro/40 | 650f / 800f |
| Apollo 1240/25 | 1150f |
| Module SCSI Apollo | 500f |
| PPC603e160+68040/25 | 1990f |
| PPC603e200+68040/25 | 2490f |
| PPC603e200+socket 60 | 2390f |
| PPC603e240+68040/25 | 3100f |
| PPC603e240+socket 60 | 2900f |
| PPC avec SCSI II | +500f |
| FlickerFixer externe tout amiga | 990f |
| Scandoubleur interne sur puces | 580f |
| **BVision PPC 8 Mo Dispo** | **1490f** |
| **Promo carte Zorro III Infinitiv** | **1990f** |
| AMIGA 4000 | |
| PPC604e200 - 233+40/25+scsi | 4190f 4790f |
| PPC604e200 - 233+socle 60+scsi | 3990f 4990f |
| Tout cable et terminaison SCSI 1,2,3 | TEL |
| Cybervision PPC 8 Mo | 1490f |
| Apollo 4040 68040/40+scsi | 1990f |
| Apollo 4060/50 ou 66+scsi | 3490f / 3890f |
| **Nous reprenons vos cartes pour l'achat d'une PPC** | |
| Carte son Delphina lite | 1690f |
| Carte son Delphina 1200 | TEL |
| DelfSer et audiolab V2 0 delphina | 300f / 699f |
| ARIADNE 2 | 750f |
| ROM 3.1 A500,A600,A2000 | 199f |
| ROM 3 1 A1200,A3000,A4000 | 299f |
| Simm 16 / 32 Mo EDD 60ns | 280f / 540f |
| Disque Dur 4 3 Go UWSCSI | 2100f |
| Disque Dur 3 2 Go IDE | 1100f |
| CD Rom 32X Plextor SCSI | 890f |
| CD Rom 36X IDE | 380f |

| DIVERS | |
|---|---|
| Lecteur 1.76Mo interne / Externe | 480f / 540f |
| Souris Wizard | 99f |
| Cable Parnet + Programmes | 100f |
| Cable 2 5/3 5/3 5 | 100f |
| Tour INFINITIV II A1200 + Alim 200W | 1300f |
| Boitier Clavier Tour Infinitiv | 349f |
| Zorro II / Zorro III | 1250f / 2650f |
| Tour A4000 | 2490f |
| Alimentation tout amiga | TEL |
| WordWorth 7 vf | 449f |
| Miami V3 ou IBrowse | 340f |
| TurboPrint 7 (va) | 369f |
| Deluxe Paint V CD | 199f |
| Scala MM400 CD (va) | 499f |
| Aminet 16 à 28 CD | 85f |
| Aminet set 2,3,4,5,6,7 CD | 199f |
| Amiga Forever 2.0 | 369f |
| Air Mail 4 1 | 50f |
| **PFS 2** | **299f** |
| **Directory Opus Magellan II** | **490f** |
| **Magellan II Mise à jour** | **260f** |
| **Dopus Plus CD** | **199f** |
| **Amiga Writer** | **499f** |
| **Art Effect 3.0** | **TEL** |
| **Tornado 3D V2.0** | **2990f** |
| **Fantastic Dreams CD (va)** | **599f** |
| **Net Connect 2 (Voyager,AmIRC,...)** | **599f** |
| **Get Connected (Miami,IBrowse,Yam)** | **499f** |
| **Candy Factory Pro CD** | **340f** |
| **Megalosound** | **299f** |
| **Zip / Jazz Tools** | **180f** |
| **Digi Booster Pro** | **240f** |
| **ProMidi Interface** | **250f** |
| **Ecran 17" SAMPO** | **2190f** |

| | |
|---|---|
| Akira CD32 ou Amiga Tools 6 CD | 80f |
| Geek Gadgets ADE 1 ou 2 CD | 100f |
| Quaterback Tools Bundle | 180f |
| Distant Suns 4 2 ou Blitz Tennis | 100f |
| Nemac IV CD ou Giga Graphics CD | 149f |
| Testament ou Master Axe ou Cygnus 8 | 129f |
| Humans 3 CD/Dk | 149f |
| Uropa 2 CD ou Magic Publisher CD | 199f |
| Shadow of 3Rd moon CD | 149f |
| Myst CD | 299f |
| Brain Damage Pinball | 149f |
| Simon Sorcerer CD | 149f |
| Megablast ou Enemy | 100f |
| Railroad Tycoon ou Colonisation | 149f |
| P-OS Pre release CD | 100f |
| Sword ou UFO ou Tiny troops | 149f |
| PPaint 7.1 CD | 249f |
| Mega Typhoon ou Theme Park | 149f |
| Abduction ou F15 Strike Eagle II | 100f |
| BlockHead 1 ou 2 | 100f |
| Skeleton Krew AGA ou Banshee AGA | 100f |
| Chaos engine 2 AGA | 149f |
| Mods Anthology CD Light Rom Gold CD | 180f |
| Light Rom 5 CD | 199f |
| Light Rom 6 CD | 290f |
| Diavolo Backup | 199f |
| Pad Competition Pro | 149f |
| **Final Odyssey CD** | **269f** |
| **Dn Escapee CD** | **269f** |
| **Samba World Cup CD ou Dk** | **269f** |
| **Quake CD** | **349f** |
| **Time Reckoning CD (Add On Quake)** | **199f** |
| **Foundation CD + MAJ** | **269f** |
| **Genetics Species CD** | **269f** |
| **Trapped 1 CD ou Trapped 2 CD** | **100f** |
| **Eat The Whistle CD** | **189f** |
| **Vulcanomogy CD** | **169f** |
| **Napalm ou Golem** | **TEL** |
| **500 Screen Savers 2 CD** | **85f** |
| **Cybergraph X 4 CD** | **169f** |
| **SuperView Productivity Suite CD** | **199f** |
| **Red Hat Linux 5.1 CD** | **129f** |
| **Siamese System V2.1 vo** | **299f** |

**PRIX NETS TTC MODIFIABLES SANS PREAVIS - CHEQUE A L'ORDRE DE MYGALE**

# Tout l'univers du PC




## Chaque mois, le magazine + 2 CD-Rom = 42 F

**Ludi CD :**
Deux exclusivités : Resident Evil 2 et Quest For Glory 5 !

**CD Pro :**
Le serveur Web Apache porté sous Windows avec ses sources






# Une interface graphique plus seyante

*Le programme de ce mois-ci est le même que le mois dernier. Il s'enrichit cependant de l'utilisation de la librairie Triton et de l'apparition d'une interface configurable, avec laquelle on peux lancer Adoom...*

Tout d'abord, nous allons initialiser le gadget "cycle". Pour celui-ci, nous avons tout d'abord créé une liste. Ensuite, partant du principe que nous sommes obligés d'y glisser l'adresse du texte, nous avons d'abord utilisé des variables de types string (S$,P020$, etc.) dans lesquelles nous avons intégré notre texte (Standard, 68020, etc.). Puis, nous plaçons l'adresse de ces variables dans chaque case de la List "cycle_gad". Pour récupérer une adresse d'une variable d'un type quelconque, il suffit d'inscrire la lettre "&" juste devant son nom.
Cette méthode est un peu lourde, mais prend moins de mémoire que son équivalent, la commande "Null()" laquelle retourne l'adresse d'un texte de la manière suivante :

```
adresse.l=Null('essai')
```

Par ailleurs, pour créer l'interface, on doit utiliser des Tags. On initialise les Taglists avec la fonction "InitTagList". Cette commande doit avoir deux paramètres : le premier est son numéro et le deuxième, sa taille. On utilise un Taglist en faisant "Use Taglist 1", où 1 est son numéro. Pour remplir ce Taglist, on utilise la commande AddTags. Un Taglist doit toujours se terminer par "AddTags #TAG_END,0" ; cette combinaison en signale la fin.

**Création de l'interface :**
Notre interface Triton doit toujours commencer de la manière suivante :

```
Use TagList 1
  AddTags
#TRCA_Name,Null('Interface
ADoom')
  AddTags
#TRCA_LongName,Null( Interface
ADoom')
  AddTags #TRCA_Info,Null('le 12/02/99')
  AddTags #TAG_END,0
```

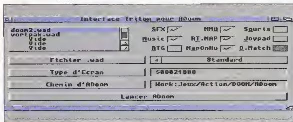
Voilà ce qui permet de définir le nom de la fenêtre et de créer par la suite l'application, avec la commande "TR_Create App_" de la triton.library. On crée l'interface grâce à des macros. La macro !HorizGroupEAC permet quant à elle de disposer les gadgets ou autres sur le plan horizontal, EAC signifiant "A part égale" (EQUALSHARE), "Aligné" et "Centré". On peut voir toutes ces macros dans le fichier triton.bb2. La différence entre une interface Triton et une interface Gadtools vient, entre autres, de la gestion des gadgets lorsque l'on change la taille de la fenêtre. C'est pour cela que lors de la création d'une interface Triton, nous n'avons pas à définir les coordonnées des gadgets : on se sert des macros *!VertGroup*, *!HorizGroup* et *!Space* pour placer les gadgets comme on le souhaite.

**Récupérer les événements des gadgets**
Pour récupérer les événements dans notre fenêtre, on se sert de listes chaînées. Afin de recueillir le numéro d'un gadget, on a recours à "*trmsg\trm_ID" Le "*" est un pointeur. Il permet de pointer sur une adresse, comme ici sur trm_ID. Attention "*trmsg" doit être de type TR_Message. On effectue cette action en écrivant "*trmsg.TR_Message". Pour rentrer en possession du numéro d'une ligne sélectionnée dans un listeview, on écrira "*trmsg\trm_Data"


*Tirs partagés.*

**Avant de coder**
Pour faire fonctionner cet exemple, vous devez sélectionnez, dans le menu Compiler, l'item "Compiler Options..." (raccourci Amiga O) et inscrire, dans la case Resident, "blitzlibs:triton.res". Le fichier triton.res et tous les autres fichiers concernant cette librairie sont disponibles sur la page Web de l'auteur (http://www chez.com/titan/SourcesBlitz2.html) ; il vous reste à copier triton.res dans blitzlibs:. De plus, vous devez copier triton.library1 dans blitzlibs:amigalibs\. Une fois ce fichier copié, lancez makedeflibs ou LibMan afin de créer un nouveau fichier deflibs (le blitz se sert de celui-ci comme d'une grosse librairie au démarrage). Pour trouver le kit complet qui permet de programmer des interfaces en Triton, téléchargez http://us.aminet net /pub/aminet/dev/gui/tri20b2dev.lha

*Benjamin Vernoux - bvernoux@kyxar.fr*



*Il n'y a aucun mal à changer de police, Triton.*

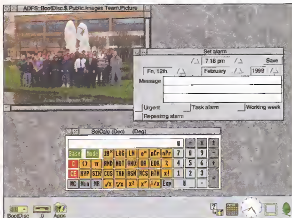
```
WBStartup
WbToScreen 0
NEWTYPE.listvi
aitem.w
bitem s
End NEWTYPE
DEFTYPE.w SFX1,MMU1,Souris1,Music1,RTMAP1,Joypad1
DEFTYPE.w RTG1,MapOnHu1,DMatch1,fichier,CPU1
MaxLen pa$=192
MaxLen fi$ 192
Dim List MaList.listview(10)
Dim List Chemin listview(10)
For i=0 To 9
     If AddItem(MaList()) Then MaList()\bitem = '    Vide
'
     If AddItem(Chemin()) Then Chemin()\bitem = ''
Next i
S$='Standard'
P020$='68020'
P030$='68030'
P040$='68040'
P060$ '68060'
Dim cycle_gad 1(6)
cycle_gad(1)=&S$
cycle_gad(2)-&P020$
cycle_gad(3)=&P030$
cycle_gad(4)-&P040$
cycle_gad(5)=&P060$
cycle_gad(6)=0
string.s=' '
InitTagList 1,20
InitTagList 2,200
Use TagList 1
  AddTags #TRCA_Name,Null('Interface ADoom')
  AddTags #TRCA_LongName,Null('Interface ADoom')
  AddTags #TRCA_Info,Null('le 12/02/99')
  AddTags #TAG_END,0
Use TagList 2 ; Interface D'ADoom
  AddTags
!WindowID{1},!WindowPosition{#TRWP_CENTERDISPLAY};,!WindowD
imensions{440,165}
  AddTags !WindowTitle{Null('Interface Triton pour
ADoom')};,!WindowFlags{#TRWF_NOSIZEGADGET}
AddTags !VertGroupEAC,!SpaceB
     AddTags !HorizGroupEAC
         AddTags !Space,!ListSS{&MaList(0)-
36,10,0,0},!Space
       AddTags !VertGroupSAC
         AddTags !HorizGroupEAC,!TextID{Null('
_SFX'),1},!SpaceS,!CheckBox{1},!EndGroup,!Space
         AddTags !HorizGroupEAC,!TextID{Null('
_Music'),4},!SpaceS,!CheckBox{4},!EndGroup,!Space
         AddTags !HorizGroupEAC,!TextID{Null('
_RTG'),7},!SpaceS,!CheckBox{7},!EndGroup
       AddTags !EndGroup,!Space
       AddTags !VertGroupSAC
         AddTags !HorizGroupEAC,!TextID{Null('
MM_U'),2},!SpaceS,!CheckBox{2},!EndGroup,!Space
         AddTags !HorizGroupEAC,!TextID{Null('
R_T.MAP'),5},!SpaceS,!CheckBox{5},!EndGroup,!Space
         AddTags
!HorizGroupEAC,!TextID{Null('M_apOnHu'),8},!SpaceS,!CheckBo
x{8},!EndGroup
       AddTags !EndGroup,!Space
       AddTags !VertGroupSAC
         AddTags !HorizGroupEAC,!TextID{Null('
S_ouris'),3},!SpaceS,!CheckBox{3},!EndGroup,!Space
         AddTags !HorizGroupEAC,!TextID{Null('
_Joypad'),6},!SpaceS,!CheckBox{6},!EndGroup,!Space
         AddTags
!HorizGroupEAC,!TextID{Null('_D.Match'),9},!SpaceS,!CheckBo
x{9},!Disabled,!EndGroup
       AddTags !EndGroup,!Space
     AddTags !EndGroup,!SpaceB
     AddTags !HorizGroupEAC,!Space
       AddTags !Button{Null('Fichier .wad'),11},!Space
       AddTags !CycleGadget{&cycle_gad(1),0,12},!Space
     AddTags !EndGroup,!SpaceB
     AddTags !HorizGroupEAC,!Space
       AddTags !Button{Null('Type d Ecran'),13},!Space
       AddTags !ClippedTextBox{&string,14},!Space
     AddTags !EndGroup,!SpaceB
     AddTags !HorizGroupEAC,!Space
       AddTags !Button{Null('Chemin d'ADoom'),15},!Space
       AddTags !ClippedTextBox{&string,16},!Space
     AddTags !EndGroup,!SpaceB
     AddTags !HorizGroupEAC,!Space
       AddTags  !Button{Null('Lancer ADoom'),17},!Space
     AddTags !EndGroup,!SpaceB
AddTags   !EndGroup,!EndProject
AddTags #TAG_END,0
Use TagList 1
application.l=TR_CreateApp_(TagList)
If application
        Use TagList 2
        project1.l=TR_OpenProject_(application,TagList)  ;
ouvre l'ecran de l'interface
    Else                                                  ;
et affiche l'ecran plus les gadgets
        TR_DeleteApp_ application
EndIf
If (project1)
      Else
      TR_DeleteApp_ application
EndIf
SFX$='-nosfx '
AA$=ProgPath$
p$=GetPathName(AA$)
prefs$=p$+'ADoom.prefs'
Gosub ChargerPrefs
Gosub Prefs
Repeat
  TR_Wait_ application,0 ; identique à WaitEvent
  *trmsg TR_Message=TR_GetMsg_(application) ; comme la
commande Event
If *trmsg
        Select *trmsg\trm_Class
                  Case #TRMS_CLOSEWINDOW ,boutton de
fermeture
                   quit=True       ; de la fenetre active
        End Select
      Select *trmsg\trm_ID ; Identique à la commande
GadgetHit
          Case 1 ;Bouton SFX
```

```
        [illegible] *trmsg\trm_Data=1 SFX$=[illegible]
lse SFX$=[illegible]
      Case [illegible] ;Bouton MMU
        If *trmsg\trm_Data=1 MMU$='-mmu ':MMU1=-1
lse MMU$=[illegible]:EndIf
      Case [illegible] ;Bouton Souris
        If *trmsg\trm_Data=1 Souris$='-mouse
':Souris1=-1 Else Souris$='':Souris1=0:EndIf
      Case 4 ;Bouton Music
        If *trmsg\trm_Data=1 Music$='-music
':Music1=-1 Else Music$='':Music1=0:EndIf
      Case 5 ;Bouton RT.MAP
        If *trmsg\trm_Data=1 RTMAP$='-rotatemap
':RTMAP1=-1 Else RTMAP$='':RTMAP1=0:EndIf
      Case 6 ;Bouton Joypad
        If *trmsg\trm_Data=1 Joypad$='-joypad
':Joypad1=-1 Else Joypad$='':Joypad1=0:EndIf
      Case 7 ;Bouton RTG
        If *trmsg\trm_Data=1 RTG$='-rtg ':RTG1=-1
Else RTG$='':RTG1=0:EndIf
      Case 8 ;Bouton MapOnHu
        If *trmsg\trm_Data=1 MapOnHu$='-maponhu
':MapOnHu1=-1 Else MapOnHu$='':MapOnHu1=0:EndIf
      Case 9 ;Bouton D.Match
        If *trmsg\trm_Data=1 DMatch$='-deathmatch
':DMatch1=-1 Else DMatch$='':DMatch1=0:EndIf
      Case 10 ;Listeview Chemin
          fichier=*trmsg\trm_Data
          fich$=' file '+Chemin(fichier)\bitem+' '
      Case 11 ;Bouton Chemin
          path$=ASLFileRequest$('Fichier
wad',pa$,fi$,'#?.wad')
        If path$
          err=FirstItem(MaList())
          err=FirstItem(Chemin())
        MaList(fichier)\bitem = fi$
        Chemin(fichier)\bitem = path$
        TR_SetAttribute_ project1,10,0,NOT 0
        ResetList MaList()
        TR_SetAttribute_ project1,10,0,&MaList(0)-36
        EndIf
      Case 12 ;CycleGadget CPU
        Select *trmsg\trm_Data
          Case 0 CPU$='':CPU1=0
          Case 1 CPU$='-cpu 68020 ':CPU1=1
          Case 2:CPU$='-cpu 68030 ':CPU1=2
          Case 3 CPU$='-cpu 68040 ':CPU1=3
          Case 4:CPU$='-cpu 68060 ':CPU1=4
        End Select
      Case 13 ;Bouton Type D Ecran
          screenmode.l=RTEZScreenModeRequest('Choix
de Mon Ecran')
          smod.l=Peek.l(screenmode)
        If screenmode<>0
          smode$='$'+Hex$(smod.l)
          text$=smode$
          TR_SetAttribute_
project1,14,#TRAT_Text,&text$
          screenmod$='-screenmode '+smode$+' '
        EndIf
      Case 15 ;Chemin D ADoom
          ADoomfile$=ASLFileRequest$('Fichier
ADoom',pa$,fi$,'#?')
          If ADoomfile$
            ADoom$=ADoomfile$
            TR_SetAttribute_
project1,16,#TRAT_Text,&ADoom$
          EndIf
      Case 17 ;Bouton Lancer ADoom
          cli1$='CON:0/50/640/150/Chargement D'ADoom en
cours.../AUTO/CLOSE'
        cli$='CON:0/0/640/50/Prefs d'ADoom/AUTO/CLOSE'
      DOOM$=ADoom$+'
'+fich$+screenmod$+SFX$+MMU$+Souris$+Music$+RTMAP$+Joypad$+R
TG$+MapOnHu$+DMatch$+CPU$+'-forcedemo'+Chr$(10)
        *fh.l=Open_(&cli$,1004)
        errorr.l=FPuts_(*fh,&DOOM$)
        *fh1.l=Open_(&cli1$,1004)
        Execute_ &DOOM$,*fh,0
        succ.l=Close_(*fh)
        succ.l=Close_(*fh1)
    End Select
EndIf
TR_ReplyMsg_ *trmsg  , on doit toujours repondre a 1
TR_GetMsg_
Until quit=True , on va a Repeat jusqu a ce que quit=-1 (ou
True)
TR_CloseProject_ project1
TR_DeleteApp_ application
Free TagList 1
Free TagList 2
Gosub SauverPrefs
End
.ChargerPrefs
If ReadFile(0,prefs$)
  FileInput 0
    SFX1=Val(Edit$(4))
    MMU1=Val(Edit$(4))
    Souris1=Val(Edit$(4))
    Music1=Val(Edit$(4))
    RTMAP1=Val(Edit$(4))
    Joypad1=Val(Edit$(4))
    RTG1=Val(Edit$(4))
    MapOnHu1=Val(Edit$(4))
    DMatch1=Val(Edit$(4))
    fichier=Val(Edit$(4)) ; fichier .wad selectionne
    CPU1=Val(Edit$(4))
    smode$=Edit$(20)       , mode d Ecran
    ADoom$=Edit$(192)      , Chemin du programme ADoom
    err=FirstItem(MaList())
    err=FirstItem(Chemin())
    MaList()\bitem = Edit$(120)
    Chemin()\bitem = Edit$(192)
  For i=0 To 9
    succ=NextItem(MaList()):MaList()\bitem = Edit$(120)
    succ=NextItem(Chemin()):Chemin()\bitem = Edit$(192)
  Next i
  CloseFile 0
EndIf
Return
.Prefs
 TR_SetAttribute_ project1,1,#TRAT_Value,SFX1:If SFX1=0
SFX$='-nosfx ':Else SFX$='':EndIf
 TR_SetAttribute_ project1,2,#TRAT_Value,MMU1:If MMU1=-1
```

```
MMU$='-mmu ' Else MMU$='' EndIf
  TR_SetAttribute_ project1,3 #TRAT_Value,Souris1:If
Souris1=-1 Souris$='-mouse ' Else Souris$='' EndIf
  TR_SetAttribute_ project1,4,#TRAT_Value,Music1:If
Music1=-1 Music$='-music ' Else Music$='':EndIf
  TR_SetAttribute_ project1,5,#TRAT_Value,RTMAP1:If
RTMAP1=-1 RTMAP$='-rotatemap ' Else RTMAP$='' EndIf
  TR_SetAttribute_ project1,6,#TRAT_Value,Joypad1:If
Joypad1=-1 Joypad$='-joypad ' Else Joypad$='':EndIf
  TR_SetAttribute_ project1,7,#TRAT_Value,RTG1:If
RTG1=-1 RTG$='-rtg ' Else RTG$='' EndIf
  TR_SetAttribute_ project1,8,#TRAT_Value,MapOnHu1 If
MapOnHu1=-1 MapOnHu$=' maponhu ' Else MapOnHu$='':EndIf
  TR_SetAttribute_ project1,10,0,NOT 0  , efface le
listview
  ResetList MaList()                  , on se remet
au début de la liste MaList()
  TR_SetAttribute_ project1,10,0,&MaList(0)-36 , on
raffraichit le ListeView avec la nouvelle Liste
  TR_SetAttribute_ project1,10,#TRAT_Value,fichier , on
met en surbrillant le fichier .wad selectionne
  TR_SetAttribute_ project1,16,#TRAT_Text,&ADoom$ ;
Affiche le chemin d Adoom
Select CPU1
      Case 0:CPU$=''
      Case 1:CPU$='-cpu 68020 '
      Case 2:CPU$='-cpu 68030 '
      Case 3:CPU$='-cpu 68040 '
      Case 4:CPU$='-cpu 68060 '
End Select
;GTSetString 1,14,Centre$(smode$,33)
screenmod$='-screenmode '+smode$+' '
TR_SetAttribute_ project1,14,#TRAT_Text,&smode$
Return
.SauverPrefs
If WriteFile(1,prefs$)
    FileOutput 1
       NPrint Str$(SFX1)
       NPrint Str$(MMU1)
       NPrint Str$(Souris1)
       NPrint Str$(Music1)
       NPrint Str$(RTMAP1)
       NPrint Str$(Joypad1)
       NPrint Str$(RTG1)
       NPrint Str$(MapOnHu1)
       NPrint Str$(DMatch1)
       NPrint Str$(fichier)  ; fichier .wad selectionne
       NPrint Str$(CPU1)
       NPrint smode$   ; mode d Ecran
       NPrint ADoom$   , Chemin du programme ADoom
       err=FirstItem(MaList())
       err=FirstItem(Chemin())
       NPrint MaList()\bitem
       NPrint Chemin()\bitem
   For i=0 To 9
      succ=NextItem(MaList()):NPrint MaList()\bitem
      succ=NextItem(Chemin()):NPrint Chemin()\bitem
   Next i
    CloseFile 0
EndIf
Return
;fin
```

# Le BBC Basic : introduction

*Un langage à tout faire. Cette nouvelle série d'articles propose l'apprentissage, à la base, du BBC Basic et du Basic en général.*



Quelques programmes écrits en Basic.

Une partie sera applicable aux autres Basic, tel que le GwBasic ou le QuickBasic, mais les spécificités du BBC Basic ne serviront qu'aux machines Acorn. Pour cette première, il convient de présenter le plan de progression de notre apprentissage.

**Présentation du BBC Basic**

Le Basic est un langage ancien, destiné avant tout à apprendre les rudiments de la programmation aux débutants. Les années lui ont valu d'évoluer pour devenir un langage procédural de haut niveau, proche du Pascal. Le BBC Basic a fait son apparition avec les BBC Modèles A et B d'Acorn, des machines 8 bits : celui-ci est le descendant de l'Atom Basic, livré avec l'Atom d'Acorn. Le BBC correspondait à une gamme d'ordinateurs novateurs, servant de support à une émission d'apprentissage à l'informatique, initiée par la BBC. L'arrivée de l'Archimedes et de ses importantes capacités en mémoire ont permis de faire passer la taille de l'interpréteur, de 16 Ko à environ 64 Ko. Cet espace supplémentaire est utilisé pour l'ajout de fonctionnalités très évoluées, qui font du BBC Basic un des plus puissants et des plus rapides Basics au monde. Les versions actuellement exploitées sont les *BBC Basic V* et *VI*. La *VI* apporte des possibilités évoluées pour le traitement des nombres flottants. Le BBC Basic dispose de fonctionnalités-clés : Basic procédural complet, incluant une gestion des librairies, accès complet aux ressources système ainsi qu'à la ligne de commande, assembleur intégré, etc. Le BBC Basic correspond à un langage interprété, ce qui signifie que chaque ligne de programme est transcrite et transformée en instructions compréhensibles par le microprocesseur, au fur et à mesure de l'exécution du programme. Un programme compilé subit ce traitement de façon préalable, ce qui le métamorphose en code directement utilisable. Un code compilé est souvent plus rapide, mais difficile à maintenir. Si l'interpréteur tombe sur une erreur, il s'arrête immédiatement en affichant un message ad hoc, ce qui facilite le débogage (la chasse aux erreurs). La vitesse de l'interpréteur de chez Acorn fait qu'un programme en BBC Basic se révèle souvent aussi rapide qu'un autre écrit en C.

**Plan**

Le BBC Basic est un langage complet ; nous l'aborderons de la manière suivante :

I. Présentation des mots-clés classiques du Basic.

II. Mode procédural du BBC Basic et librairies.

III. Spécificités du BBC Basic (calculs complexes, accès évolué aux blocs de données, etc.).

VI. Accès à la ligne de commande, aux interruptions logicielles et commandes Fx.

IV. Fonctions graphiques, commandes Vdu et mode Teletext.

V. Contrôle du son.

VI. Optimisation des routines via l'assembleur intégré.

VII. La programmation sous *Wimp*.

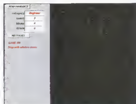
On trouvera aussi d'autres séries d'articles qui s'intercaleront dans les parties principales, comme par exemple :

Maîtrise de la GameSuite (fin de la partie V).

Compilateurs Basic (début de la partie VI).

Interactions Basic/Assembleur (fin de la partie VI).

Utilisation de *MenuUtils* et de *DrWimp* (partie VII).

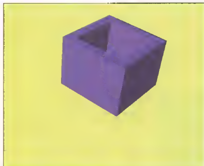


Un jeu sous Wimp, 100 % en Basic.

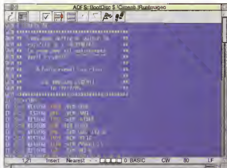
Chaque article s'accompagnera de nombreux exemples sur le CD : la simplicité du Basic favorise l'apprentissage par l'exemple. Il va sans dire que faire le tour du sujet prendra un temps considérable. On peut considérer le BBC Basic comme la boîte à outils ultime pour programmer sous RiscOs : aucun programme n'est irréalisable en BBC Basic. Chacun peut y trouver son intérêt, du programmeur amateur au chercheur désireux de se servir du langage comme base pour des routines plus complexes, en assembleur par exemple.

**Comment l'utiliser ?**

Tous les Basic ne se valent pas : le BBC Basic est l'un des plus évolués, mais n'existe que sur machines Acorn. Un ordinateur Acorn 8 bits ne permettra pas de profiter des évolutions des *BBC Basic V* et *VI*. En revanche, un Archimedes constitue un achat raisonnable pour qui veut se faire plaisir : difficile à trouver, il ne coûte pas cher et apporte une puissance équivalente à un *Amiga 1200* (s'il se



*De la 3D en Basic.*




*Le code du programme précédent.*

trouve muni d'un *Arm2*) ou à un *486SX* (s'il est équipé d'un *Arm3*). Si votre machine sert à la création de petits programmes en Basic, vous aurez peut-être besoin de démarrer la machine directement avec l'interpréteur. Pour ce faire, utilisez la commande *Configure Language 16, pour RiscOs 3.10 (pour les autres systèmes, vérifiez le numéro du module Basic avec la commande *RomModules). Un démarrage standard restera probablement tel quel, mais un démarrage avec la touche Shift pressée vous amènera directement dans le Basic.

Le mode Basic le plus simple est le mode direct. Il suffit de lancer la commande *Basic à partir de la ligne de commande de *RiscOs* afin de se trouver en mesure de l'exploiter. Le Basic exige que toutes les commandes soient en majuscules. Par exemple, PRINT 10*20 vous renverra la valeur 200 à l'écran. Toutes les commandes ne peuvent être employées en mode direct ; par exemple, GOTO, qui crée une boucle vers une ligne du programme, ne s'utilisera qu'à l'intérieur de celui-ci. Le mode direct constitue cependant un environnement pratique pour effectuer des calculs ou consulter l'aide en ligne. Pour quitter l'interpréteur, il faut taper QUIT. Si vous composez de petits programmes, vous aurez moyen de les exécuter avec la commande RUN du Basic. L'idéal consiste toutefois à exploiter un éditeur de texte apte à gérer le mode Basic. Un des plus anciens s'appelle *Arm8E* et le plus classique se nomme *Edit* ; celui-ci est livré avec chaque machine Acorn. Pour plus de confort, il existe deux choix : *Zap*, d'une puissance incomparable, et *StrongEd*, très ergonomique. Au risque de nous faire des ennemis, nous conseillons l'usage de *StrongEd*, ou à défaut, nous préconisons *Edit*, deux outils simples pour les débutants.

**Premiers pas**

Le Basic est un langage trop mal considéré de nos jours ; cela provient en grande partie de l'amalgame que certains font entre langage pour débutants et langage simpliste. Il est pourtant très souple. Voici un exemple simple :

```
10 PRINT 'Bonjour'
```




*Encore un jeu en Basic.*

Ce programme affiche la chaîne Bonjour! sur la console (comprenez à l'écran ou dans une fenêtre de l'environnement graphique). Un autre exemple :

```
10 PRINT 'Arg'
20 GOTO 10
```

Celui-ci affichera la chaîne de caractères Arg un nombre illimité de fois (le programme boucle sur lui-même).

```
10 INPUT 'Quel est ton âge',A%
20 PRINT 'Dans 10 ans tu auras ';A%+10
```

La première ligne demande l'âge de la personne et stocke la valeur dans une variable de nom A% et de type entier. La seconde ligne du programme affiche une chaîne de caractères, puis le résultat de A%+10 (A%=âge). Le Basic se compose de mots-clés qui effectuent des traitements sur l'information, comme par exemple la commande PRINT. Le programme est numéroté, bien que cela ne se révèle pas obligatoire, comme nous le verrons plus tard. Le concept de variable se montre un petit peu ambigu : il s'agit en fait d'un nom auquel on attribue une valeur. Cette variable peut contenir un élément susceptible de se transformer, comme celle de la dernière touche tapée au clavier, ou une constante comme celle de PI. Il faut néanmoins bien comprendre le point suivant : le terme "variable" désigne en fait un élément dont le contenu est soumis à d'éventuelles variations. Celles-ci n'ont rien de systématique.

**Conclusion**

Ces exemples sont simples et permettent d'aborder facilement le Basic. La maîtrise de ce langage vous donnera le contrôle de votre ordinateur : attention cependant aux idées reçues... Le Basic, aussi permissif soit-il, ne pardonne pas la fantaisie ou l'approximation. De nombreuses personnes ont affirmé que la maîtrise des langages de programmation pouvait se passer de connaissances mathématiques : c'est faux ! Un esprit logique et des bases en mathématiques restent nécessaires. Toutefois, cet acquis de départ demeure relativement modeste ; là réside le principal intérêt du Basic. En se mettant à la portée des plus jeunes, on ne s'étonnera donc pas qu'il suscite encore bien des vocations (qui n'a pas commencé avec le Basic ?).

*David Feugey*

# Internet, comment ça marche ?

*Le genre humain a réussi cette étonnante prouesse permettre à tous les ordinateurs de la planète de communiquer entre eux sur le même réseau. Toutefois, on peut se demander comment ils font pour s'orienter dans cette immense toile d'araignée.*

En 1969, l'ARPA (Advanced Research Projects Agency) crée Arpanet, un réseau expérimental destiné à étudier un système fiable de transmission de données. Le projet est un succès. Bon nombre d'organismes commencent dès les années 70 à utiliser ce réseau, qui passera du stade expérimental au stade opérationnel en 1975. Commence alors le développement d'un protocole (charte qui définit la façon dont les données doivent être échangées), qui saisira bientôt toutes les attentions : TCP/IP. En 1983, l'armée elle-même l'adopte ; il est d'ailleurs expressément demandé que la pile TCP/IP fasse partie de l'*Unix* de Berkeley, le fameux BSD. Débutait alors entre *Unix* et TCP/IP une histoire d'amour qui ne s'achèverait pas de sitôt.

Jusqu'en 1990, d'autres réseaux élaborés sur le principe de TCP/IP s'ajoutèrent à Arpanet, lequel cessa d'exister en 1995. Aujourd'hui, le nombre de réseaux qui reposent sur TCP/IP dans le monde entier est estimé à une centaine de milliers ; cela représente bien plus que ce qu'avaient pu imaginer les créateurs initiaux du réseau Arpanet. Internet ne se limite pas à un nouveau réseau : il offre une synthèse de tous ces réseaux. Par ailleurs, TCP/IP s'apprête à détrôner les géants d'hier au sein même des réseaux d'entreprise, car non seulement ce surdoué n'a pas son égal sur le réseau planétaire, mais en plus, il s'avère diablement efficace, et ce pour n'importe quelle topologie. Mais pourquoi est-il si puissant ?...

**OSI, un principe de base**

TCP/IP, comme nombre de systèmes de communications qui n'ont pas connu sa gloire, s'appuie sur une réflexion générique issue des ingénieurs de L'International Standard Organisation (ISO), lesquels ont créé un modèle, l'Open System Interconnect (ou OSI, à ne pas confondre). Celui-ci fait office de référence en matière de communication réseau. Il possède en effet cet avantage d'être adaptable à tout type de système de communication. Le voici donc détaillé :

**1. La couche physique**

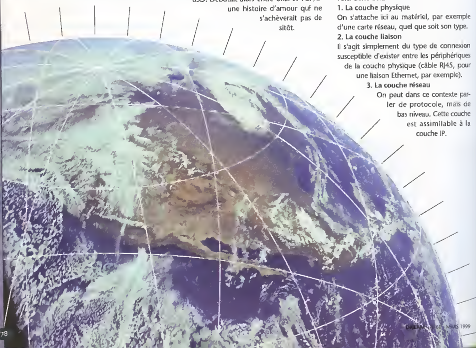
On s'attache ici au matériel, par exemple d'une carte réseau, quel que soit son type.

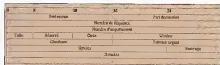
**2. La couche liaison**

Il s'agit simplement du type de connexion susceptible d'exister entre les périphériques de la couche physique (câble RJ45, pour une liaison Ethernet, par exemple).

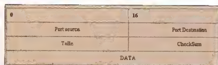
**3. La couche réseau**

On peut dans ce contexte parler de protocole, mais de bas niveau. Cette couche est assimilable à la couche IP.





*Format d'un segment TCP.*



*Format d'un segment UDP.*

**4. La couche Transport**

A partir de cet instant, l'information transite réellement et le contrôle d'erreur s'effectue. TCP s'acquitte entre autres de cette tâche.

**5. La couche Session**

Cette couche gère les connexions entre les applications et abrite les principes d'ouverture de «socket» et de «ports» du TCP/IP .

**6. La couche Présentation**

Il n'y a pas de réel équivalent de cette couche dans le monde TCP/IP. En effet, la «présentation» d'un fichier est effectuée par l'application elle-même ; lorsque par exemple on visionne une page Html dans un navigateur, celui-ci identifiera le type de document auquel il a affaire.

**7. La couche Application**

Les applications qui vont utiliser le réseau pour communiquer résident ici. C'est à ce niveau que se trouvent Ftp, Telnet, Web, etc.

Le fonctionnement de TCP/IP, bien qu'adaptable au modèle OSI universel, est quelque peu simplifié. Il peut se résumer par les quatre couches suivantes :

**1. Couche réseau**

Prend en charge les couches physique et liaison du modèle OSI. On n'aura presque jamais à s'occuper de cette partie du fonctionnement, exception faite d'une future présentation du protocole ARP.

**2. Couche Internet**

C'est ici que se situe IP, le fondement même du protocole. S'opèrent dans cette couche :

La création du datagramme (la plus petite unité de transmission)

Le système d'adressage

Le dialogue entre la couche réseau et la couche transport

Le routage des paquets

La fragmentation et la reconstruction des paquets

A noter que c'est également dans cette couche que séjourne ICMP, lequel permet d'envoyer des informations de contrôle à l'adresse source d'un paquet IP. De cette façon, les tables de routage se mettent à jour ; toutefois, ce protocole est plus connu grâce à «ping» ou encore à «traceroute».

**3. Couche transport**

Elle représente la chasse gardée des protocoles UDP et TCP. Il est important de bien comprendre la différence entre UDP et TCP. Ces protocoles, bien que logés tout deux dans la couche transport, ne fonctionnent pas de la même manière. On a coutume de dire que TCP est plus «sécurisé» qu'UDP. En effet, UDP fonctionne en mode «non connecté» ; cela signifie qu'il n'effectue aucun contrôle sur le bon arrivage des données qu'il a fait transiter. TCP, quant à lui, fonctionne en «mode connecté». Il vérifie, par un système d'acquittement, que les paquets envoyés ont bien été reçus. Ce système porte aussi le nom de «three way handshake». Prenons un exemple pour illustrer cette technique.

Une machine X souhaite transférer des données vers une machine Y. Elle commence par lui envoyer un paquet marqué du bit «SYN» (synchronisation) ; la machine Y le reçoit et renvoie à X un paquet marqué des bits «SYN» et «ACK» (acquittement). Quand X réceptionne ce paquet, elle a alors la certitude que Y est fonctionnelle et prête à accueillir des données. La transmission de données s'achève lorsque le bit «FIN» est positionné.

**4. Couche Application**

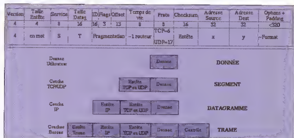
Là séjournent les applications qui utilisent le protocole TCP/IP.

**Ce n'est pas fini...**

Nous n'avons fait que lever légèrement le voile sur le fonctionnement d'Internet. Nous continuerons dans les prochains mois à progresser dans les méandres des couches IP (Internet Protocol) puis TCP (Transmission Control Protocol) et UDP (User Datagram Protocol), étudierons quelques exemples types d'applications fondées sur ces protocoles et finirons par une initiation à la programmation des applications client/serveur.

A n'en pas douter, cette première partie de l'initiation au TCP/IP n'était pas des plus exaltantes, mais ce passage obligé s'avère nécessaire pour avoir une bonne compréhension générale des applications à venir. Dans notre prochain article, nous verrons la manière dont les informations transitent par l'entremise des passerelles, et baignerons dans les joies des adresses IP et du routage...

Emile HEITOR <imil@cie.fr>



| Version | Taille Entête | Service | Taille Datag | ID | Flags | Offset | Temps de vie | Proto | Checksum | Adresse Source | Adresse Dest | Options + Padding |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 4 | 8 | 16 | 16 | 3 | 13 | 8 | 8 | 16 | 32 | 32 | <320 |
| 4 | en mot | S | T | Fragmentation | | | -1 routeur | TCP=6 UDP=17 | Entête | x | y | ~Format |


*Datagramme IP.*

### TCP/IP, un protocole... libre !

***Il est intéressant de constater comme la philosophie même de ce protocole rejoint de manière troublante le mouvement actuel, qui tend vers l'ouverture de l'information. En effet, dès sa création, TCP/IP apparut comme un standard ouvert, dont les caractéristiques furent librement publiées sur Internet, ou du moins son ancêtre. Ces publications se nomment des RFC (Request For Comment) et divulguent toutes les informations, spécifications et normalisations en rapport avec le protocole TCP/IP. Encore aujourd'hui, on peut, à sa guise, les télécharger sur Internet.***

PETITES ANNONCES

### Ventes

VDS CARTE BLIZZARD 1230 8 M. Prix : 1100 francs. Tél : 01 47 49 02 48 (répondeur).

Département 92 - vds A1200 + hd 120 Mo + disks + imprimante : 1300 francs à débattre. Tél : 01 46 05 08 62.

Département 59 : vds a1200 + mon + bliz 50 MHz 28 Mo + dd 640 Mo + l. cdr + n6x rev-cd – dk.2600 francs a500 + ext + joy + rev + jeux + edu 3 à 10 ans + util – 600 francs. Tél : 03 20 32 45 18 ap 18h.

Vds genlock yc Amiga avec cable + logiciel 1400 francs frais de port inclus. Dominique : 04 79 37 85 04 ap. 19H.

Vds A1200 + 1240 + 16 Mo + CD 6x + 1.2go + carte graph + tower + clav pc + moniteur – 4000 francs Tower A1200 + int clav + alim 250w – 900 francs. Blizzard 1230 + 16 Mo – 650 francs matériel divers. Contacter Sébastien – Auxerre (89) au 06 85 09 72 14.

Vds A1200 tour ateo + bliz 1260 16 Mo + idemux + clavier PC + écran 1085s + dd120 Mo : 4000 francs. 32 Mo edo : 400 francs. Tél : 04 90 53 41 61.

Département 30 : A1200 + pup603/160 + 40/25 + 64 Mo = 2800 francs ; blizz30/40 + fpu + 8 Mo = 650 francs ; hd 200 Mo//1.2 Mo//blizz20/ 28 MHz + 4 Mo + fpu ; dispo + débat. Tél : 04 67 82 46 01.

Vds station Alpha 255 digital. Matenel neuf. Faire offre. Tél : 01 39 51 40 53.

Département 76 : vds Digitaliseur de son DSS8 + : 200 francs. Imprimante jet d'encre Commodore MPS 1270A noir et blanc : 300 francs. Wordworth 6(vf) DK + Doc : 290 francs. Wordworth 7(vf) CD : 390 francs. Tél : 02 35 91 33 13.

Département 57 : vds carte Blizzard 1260 + 4 Mo. Prix : 1800 francs. Tél : 03-87-52-17-60 ou 06-14-56-56-56.

**Département 26 : vds Pup 603e+/240&040/25 + 64 Mo 2500 francs, tour atéo 400 francs, CD32X 200 francs, CM + Alim A1200 500 francs. Anthony au 04 75 31 06 40 le samedi.**

Département 94 : vds A1200 plusieurs conf possibles ap 18H. Tél : 01 48 93 85 99.

Département 71 : vds A4000/ 040 16/220 CD 8X Clavier Neuf à 4600 francs A1200 + 030 + 8 Mo : 1400 francs 1083 S à 250 francs. Arnaud le soir. Tél : 03 85 33 85 12.

Département 68 : vends barette mémoire SIMM 8 Mo EDO + disque dur ide 3-5P 130 Mo. Prix à débattre. Demander Marc au 03 89 06 23 30 (le soir).

Département 74 : vds Amiga 1200 Tour Micronik + 68030 à 42 Mhz + 68882 + 8 Mo + DD 250 Mo : 2200 francs. Tél : 04-50-01-27-04. Demander Jérôme.

**Département 13 : vds carte SCSI pour Blizzard 1230, 1240 ou 1260 avec câblage. Prix à débattre. Tél : 06 12 24 63 75.**

Alpha station Multia 233-300 MHz 48 Mb/16b SCSI 2 CD imprimante écran ethernet Doc 600 Euro Belgique. Tél : 00 32 (0) 10 65 68 85.

### ACHATS

Achète d'occasion le jeu LES SETTLERS ; faire offre à Luc aux heures des repas au 02 98 41 68 05.

Achète jeux pour Amiga 1200. Tél. au 01 48 34 21 12 ou Wattiez Valerie, 110 rue Henri 8arbusse 93300 Aubervilliers.

**Tarif des petites annonces**

❑ *Je suis abonné à Dream ............................................ 0F*
❑ *Je ne suis pas abonné à Dream ....................................+ 20 F*
❑ *Je choisis l'option de parution dans tous les magazines Posse (PC Team, Studio Multimédia) ........+20 F*
❑ *Je choisis l'option caractères gras ................................+20 F*
*Cochez vos choix* *Soit un total de .........*

***Rubriques :** ❑ achats ❑ ventes ❑ contacts - Département : ......*

*Règlement par chèque ou mandat à l'ordre de Posse Press/PA, 16, rue de la Fontaine au Roi, 75011 Paris. Parution de l'annonce sous un ou deux mois en fonction des dates de bouclages du magazine. La vente et l'échange de logiciels concerne uniquement les originaux. Posse Press se réserve le droit de refuser toute annonce portant à confusion.*

**Dream** est édité par Posse Press
SARL au capital de 250 000 francs
**Associés principaux** Romain Canonge, Christine Robert
**Représentant légal** Romain Canonge
16 rue de la Fontaine au Roi
75011 Paris
Tél. 01 53 36 84 10
Fax 01 43 55 66 48
Web www.posse-press.com/dream
E-mail dream@posse-press.com
**Rédaction**
**Directrice des Rédactions**
Christine Robert (84 24)
crobert@posse-press.com
**Directeur adjoint des rédactions**
Yann Serra (84 15)
yserra@posse-press.com
**Chefs de rubrique**
Karine Mondal (84 13)
kmondal@posse-press.com
David Feugey (84 13)
dfeugey@posse-press.com
**Secrétaire de rédaction**
Bernard Medioni (84 11)
**Collaborateurs permanents de la rédaction**
Christophe Kirsch Bruno Mathieu
Christophe Noël Pascal Pambrun
**Ont également collaboré**
Yannick Cadin Emile Heitec
Christophe Le Cannelier,
Christophe Morvant, Vincent Onilia
Frédéric Peschanski, Benjamin Vernoux,
Jakub Zimmermann
**Illustrateurs**
Christophe Barilen Weg
**Conception CD Rom**
**Directeur des projets multimédia**
Frédéric Botton (84 12)
fbotton@posse-press.com
**Création**
**Directeur artistique**
Pierre-Yves Roudy (84 28)
pyroudy@posse-press.com
**Rédactrices graphistes**
Julie Charvet (84 28)
crea@posse-press.com
Audrey Simon (84 27)
crea@posse-press.com
**Publicité et promotion**
**Directrice promotion et publicité**
Hélène Blondel (84 20)
hblondel@posse-press.com
**Chef de publicité**
Marie-Hélène Dusottier (84 21)
pub@posse-press.com
**Comptabilité et administration**
**Directrice comptable et administrative**
Angélina Parmentier (84 23)
**Assistante comptable et administrative**
Valérie Ambroon (84 22)
**Abonnements et vente d'anciens numéros**
**Responsable abonnements et VPC**
Thibault Béchat (84 18)
admin@posse-press.com
**Assistante Abonnements et VPC**
Michèle Fontinura (84 10)
**Fabrication**
**Photogravure** PPE (Paris)
**Impression** Léonce Deprez (Barlin)
**Duplication CD-Rom** CD Maker (Emerainville)
**Distribution** ventes
**Diffusion** MLP
**Réglages et modifications** Denis Rozes/D stimedias (05 61 43 49 89)
**Dépôt légal** à parution
**Commission paritaire** 75364
ISSN 1259-1165
**Directeur de Publication**
Romain Canonge (84 25)
rcanonge@posse-press.com

Le Groupe Posse édite également PC Team, PC Team Hors-Série et Le Studio Multimédia

POSSE. PRESS

# Abonnement

**Offre spéciale Le système Linux !**

LE SYSTÈME LINUX

**11 numéros de Dream avec un CD-Rom offert chaque mois = 429 francs**
**+ Le système Linux, manuel français, Editions O'Reilly = 280 francs**
**Soit un total de ~~709~~ francs**

# 459 francs

**Soit plus de 35 % d'économie !**

*Ouvrage également disponible en librairie*

## C O U P O N - R É P O N S E

**Oui, je souhaite m'abonner à Dream pour un an.**

❑ **Formule 1 :** abonnement d'un an à Dream avec 1 CD-Rom + Le système Linux, 459 francs à l'ordre de Posse Press. Le livre me parviendra séparément, en recommandé.

❑ **Formule 2 :** abonnement d'un an à Dream avec un CD-Rom offert chaque mois, 319 francs.

Nom : .................................. Prénom : .................................... Adresse : ..............................................................

Tél : .................................. Code postal : .................................. Ville : ..............................................................

Machine/système (facultatif) : ❑ Amiga/AmigaDos ❑ Amiga/Linux ❑ Pc/Linux ❑ Pc/Divers ❑ Be/BeOs ❑ Autre machine/autre système : ....................

Ci-joint un règlement de ....... francs, par ❑ chèque bancaire, ❑ mandat postal **à l'ordre de Posse Press.**

❑ carte bancaire : n° CB ⎵⎵⎵⎵ ⎵⎵⎵⎵ ⎵⎵⎵⎵ ⎵⎵⎵⎵ expire le ⎵⎵⎵⎵

En cas de paiement par carte bancaire, vous pouvez aussi commander :

Par téléphone : 01 53 36 84 10 du lundi au vendredi de 9h30 à 18h00

Par fax : 01 43 55 66 68, 7 jours sur 7, 24 heures sur 24

Offre à renvoyer à Dream/Abonnements, 16, rue de la Fontaine au Roi, 75011 Paris.

**DREAM**

*Tarif pour la France métropolitaine uniquement. DOM/TOM et étranger, ajouter 100 francs au prix de l'abonnement, paiement par mandat uniquement. Offre valable jusqu'au 15 Avril 1999 (cachet de la poste faisant foi)*

*Informatiques et libertés : en application de l'article L27 de la loi du 6 janvier 1978, relative à l'informatique et aux libertés, vous disposez d'un droit d'accès et de rectification pour toute information vous concernant en vous adressant à notre siège social. Les informations requises sont nécessaires à l'établissement de votre commande. Elles pourront également être cédées à des organismes extérieurs, sauf indication contraire de votre part*

# Dream's

**Le PC sans se planter**
[illegible]



**Le PC sans fausses notes**
[illegible]



**Le PC dans la poche**
[illegible]



**Toute l'histoire de la micro**
[illegible]



**Devenir imbattable aux jeux vidéo**
[illegible]



## LES ANCIENS NUMÉROS AVEC DISQUETTE AMIGA

**35F**

**Fin de stocks de logiciels Amiga**
De 30 à 100 francs pièce (Genetic Species, Myst, Cisco Heat...) Appelez au 01 53 36 84 10 pour connaître les disponibilités et réserver.



• **Dream n°1**
Dossier : la CD-32
Reportage : l'ECTS de Londres
Disquette : DSS 1, version complète
• **Dream n°2**
Dossier : les jeux de Noel
Reportage : comparatif numériseurs
Disquette : Amos Turbo
• **Dream n°3**
Dossier : les imprimantes
Reportage : le Supergames show
Disquette : Cinemorph, version complète
• **Dream n°4**
Dossier : les virus
Reportage : le Winter Ces de Las Vegas
Disquette : Quickwrite, version complète
• **Dream n°5**
Dossier : sex'n'Amiga
Labo : Blitz Basic 2, Mediapoint...
Disquette : Skidmarks démo
• **Dream n°6**
Dossier : le salon Imagina 94
Labo : ADPro 2 5, Quarterback...
Disquette : Personal Paint 4, version complete
• **Dream n°7**
Dossier : le piratage
Labo : Scenery Animator, Vidi 24, le PAR ..
Disquette : Canon Studio, version complète
• **Dream n°8**
Dossier : les simulateurs
Labo : Overdrive CD, Caligari, Directory Opus...
Disquette : Personal Font Maker, version light
• **Dream n°9**
Dossier : l'Amiga en vacances
Labo : Imagine 3 0, Emplant
Disquette : Abank
• **Dream n°10**
Dossier : les mangas et l'Amiga
Labo : carte M-Tec, Can do ..
Disquette : une sélection de DP
• **Dream n°11**
Dossier : l'avenir de l'Amiga
Labo : Lightwave, Dice, Videostage ..
Disquette : Painter 3D, version complète
• **Dream n°12**
Dossier : la création d'un jeu
Labo : DSS 3.0, DirWork, Brilliance 2...
Disquette : Desktop Magic
• **Dream n°13**
Dossier : l'Amiga à la télé
Labo : PageStream, Panorama
Disquette : Virus Checker 6.41
• **Dream n°14**
Dossier : l'Amiga et les graphismes
Reportage : le World of Amiga
Disquette : Real Time Sound Processor II Lite
• **Dream n°15**
Dossier : le rachat de l'Amiga
Labo : Photogenics...
Disquette : Skidmarks II, la démo
• **Dream n°16**
Dossier : les clones Amiga arrivent
Reportage : l'Amiga s'impose à Imagina
Disquette : Photogenics, la démo
• **Dream n°17**
Dossier : la Doom-mania arrive sur Amiga
Labo : GoldEd 2.1.1, Easy Calc, Studio Professional 2..
Disquette : DSS 3, version complète
• **Dream n°18**
Dossier : l'ECTS de Londres
Labo : DPaint V, Scala mm400, Sherlock, Diavolo, Power CD...
Disquette : Brutal, la démo
• **Dream n°19**
Dossier : les dessous du rachat par Escom
Reportage : l'Amiga à L'E3 de Los Angeles
Disquette : OctaMed pro 5.04, version complète
• **Dream n°20**
Dossier : l'Amiga et Internet
Labo : Photogenics 1.2, Kit Tower 1200, Digita Organiser...
Disquette : Obsession Flipper, la démo
• **Dream n°21**
Dossier : transformez votre Amiga en PC, Mac...
Labo : OctaMed 6, AsimCDFS, Fax on ...
Disquette : Flight Of The Amazon Queen, la démo
• **Dream n°22**
Dossier : l'avenir ludique de l'Amiga
Labo : CD Write, lecteur Zip.
Disquette : Vroom Multiplayer, version complète
• **Dream n°23**
Dossier : optimisez votre Workbench !
Reportage : l'Amiga aux commandes du virtuel
Disquette : Coala, la démo
• **Dream n°24**
Dossier : choisir un lecteur CD-Rom pour Amiga
Labo : Devpac 3 5, carte Piccolo, Action replay 1200 ..
Disquette : Boston Bomb Club, version complète
• **Dream n°25**
Dossier : l'Amiga expo fait un carton
Labo : Directory Opus 5.0
Disquette : Virus Checker 8 03
• **Dream n°26**
Dossier : les ordinateurs exotiques
Labo : ImagineVision, Blitz Basic 2 1, Arts et frontières..
Disquette : Zeewolf 2, la démo
• **Dream n°27**
Dossier : le guide du programmeur
Labo : la carte 68060 Blizzard 1260T
Disquette : Gloom Deluxe, la démo
• **Dream n°28**
Dossier : les extension du PC au service de l'Amiga
Labo : Tower TBD, SX32, TurboCalc 3 5, @-Net...
Disquette : TurboPrint Professional 4 1, version light
• **Dream n°29**
Dossier : Viscorp rachète l'Amiga
Labo : DeltaTower, DiskSalv 4 ..
Disquette : MandelTour AGA, version complète
• **Dream n°30**
Dossier : toutes les clés pour éviter la galère
Labo : Wordworth 5, World Atlas, encyclopédie Adfi ..
Disquette : Blitzbank 2.1
• **Dream n°31**
Dossier : le quizz de l'été
Labo : Image FX 2 a, Artpad 2, Vid 24, Kit Internet...
Disquette : une sélection de DP
• **Dream n°32**
Dossier : quel fournisseur Internet choisir ?
Labo : Turbo print v 4.1..
Disquette : Powerolds
• **Dream n°33**
Dossier : le multimédia et l'Amiga
Labo : la carte Cyberstorm 060
Disquette : une sélection de DP
• **Dream n°34**
Dossier : la communauté Amiga à la rescousse
Labo : OctaMED, Soundstudio, Imagine 5.0...
Disquette : Alien Breed 3D 2, la démo
• **Dream n°35**
Jeu : Capital Punishment
Labo : iBrowse, InterOffice, MaxDOS
• **Dream n°36**
Dossier : pourquoi investir sur Amiga
Labo : 2 modems 33600bps, X-DVE, SMD-100...
Disquette : une sélection de DP
• **Dream n°37**
Dossier : gagnez de l'argent avec votre Amiga
Reportage : l'histoire de l'informatique à Londres
Disquette : Magic Menu 2 15
• **Dream n°38**
Amigazone : Typesmith, Wordworth
Linuxzone : Lincity, Xquad, Lyx
Disquette Amiga (épuisée)

# bootik

LES ANCIENS NUMÉROS AVEC CD AMIGA, ATARI, BE, LINUX, RISC PC...

35F

• **Dream n°39**
Amigazone : les browsers Web, Shapeshifter...
Linuxzone : XWindow, The Gimp, Mesa OpenGL, Xclamation...

• **Dream n°40**
Amigazone : Blobz, Cybervision 64 3D, Armail
Linux : les distributions, Maelstrom, DDD, CDE...

• **Dream n°41** (épuisé)

• **Dream n°42**
Pourquoi Java fait peur à Microsoft ?
POS décortiqué
X-Window : installation pas à pas

• **Dream n°43**
Linux enfin expliqué aux débutants
Atari : comment installer Linux et X-Window
Risc Pc : quel kit internet choisir ?

• **Dream n°44** (épuisé)

• **Dream n°45**
Dossier : Latex pour tous les systèmes
Labo : Art effect 2, Applixware 4.3...
Plein feux : les processeurs Alpha

• **Dream n°46** (épuisé)

• **Dream N°47** (épuisé)

• **Dream N°48**
Dossier : à la découverte des ordinateurs de poche
Labo : la Redhat 5.0 est arrivée, tous les OS de l'Atari, le Syjet sur Amiga, installation d'Os/2...
Sur le CD : tout Linux pour PowerMac et la version exclusive de Tornado 3D.

• **Dream N°49**
Comment Netscape veut s'emparer d'Internet !
Puces en cuivre : Ibm pousse le PowerPc à 1 Ghz
Mise au point : que vaudra Windows Nt 5 ?
S.u.s.e. : la distribution Linux ultime ?

• **Dream N°50**
Linux, Amiga, BeOS : comment fonctionne un système d'exploitation ?
Rhapsody, le croisement parfait entre Unix et MacOS !
La compilation du noyau enfin expliquée !
Wordperfect 7 et Wordworth 7, les traitements de textes alternatifs absolus

• **Dream N°51**
Dossier : comment choisir du matériel alternatif ?
Connexion Internet : pourra-t-on aller plus vite ?
Périphériques Scsi : toutes les normes enfin expliquées !

• **Dream N°52**
Apple, Amiga, Acorn... Toutes les machines de la rentrée en avant-première !
Test : la nouvelle interface de Linux s'appelle Gnome !
Spécial : Windows 95, Linux, Amiga... Tous en réseau !
Emacs : le compagnon essentiel de Linux enfin décortiqué !

• **Dream N°53**
Micros, consoles, stations : toutes les nouvelles puces 3D au service de l'imaginaire ! A quoi sert vraiment Java ?
Test : éxécuter des applications Windows 95 sous Linux

• **Dream N°54**
Dossier : les secrets de l'informatique théorique révélés
Format MP3 : comment fonctionne-t-il ?
Comment booster son Amiga ?
Scandale : la sécurité de Windows ébranlée par un grain de sable !

• **Dream N°55**
Dossier : les réseaux de neurones
Clones 3D : la visio-conférence enfin en temps réel !
Guide d'achat des stations Alpha
Pratique : installer un lecteur SparQ sous Linux
L'an 2000 sera-t-il Unix ou NT ?

• **Dream N°56**
Dossier : pourrez-vous devenir provider Internet ?
Carte accélératrice : la Centurbo II
Culture : comment est né le phénomène Linux ?
Linux PPC : toutes les astuces pour bien l'installer

• **Dream N°57**
Dossier : les cartes à puce au secours de la sécurité informatique
En pratique : Installer Linux sur une station Alpha
Nétiquette : Newsgroups, mode d'emploi
Système : BeOs release 4 est arrivé !

• **Dream N°58**
Dans le magazine et sur le CD-Rom : tout pour faire de la 3D sous Linux
Test de la RedHat 5.2. Comment trouver des stations Alpha à bas prix ?

• **Dream N°59**
Bricoler un supercalculateur avec des vieux Pc
Bien choisir son langage de programmation
Les fonctions cachées de Lilo

**199 F Space Station Simulator**

Le simulateur de Station Spatiale en 3D temps Réel. A partir d'une conception originale de la NASA et de l'ESA. Un CD interactif exceptionnel présentant une base de données scientifique incomparable sur les sites de lancement et les modules construits pour ce projet de station internationale. Fonctionne sous Windows uniquement (désolé...)

Introduction to the International Space Station — esa — NASA — NASDA — Canada

## B O N  D E  C O M M A N D E

Coupon à renvoyer à Posse Press Dream's bootik : 16, rue de la Fontaine au Roi, 75011 Paris.

Code client (facultatif) : ........................
Nom : ........................
Prénom : ........................
Adresse : ........................
........................
Code postal : ........................
Ville : ........................
Téléphone (indispensable en cas de rupture de stock) : ........................
Machine/Système : ........................

| Articles choisis | Prix |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| Forfait port : 25 francs (France et Dom), 60 francs (étranger et Tom) | + port |
| Total | |

DREAM n°60

Je règle par :
❑ chèque à l'ordre de Posse Press ❑ mandat ❑ carte bancaire n° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_| expire le |_|_|_|_|
Date et signature obligatoires :

En cas de paiement par carte bancaire, vous pouvez aussi commander :
• Par téléphone : 01 53 36 84 10 du lundi au vendredi de 10h30 à 18h00
• Par fax : 01 43 55 66 68, 7 jours sur 7, 24 heures sur 24